Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Extração de características de RNAs não-codificadores longos utilizando o algoritmo Random Forest

Daniel Dantas Nascimento dos Santos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Engenharia de Computação

Orientadora
Prof.ª Dr.ª Maria Emilia M. T. Walter

Brasília
2016

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Engenharia de Computação

Coordenador: Prof. Dr. Ricardo Pezzuol Jacobi

Banca examinadora composta por:

Prof.ª Dr.ª Maria Emilia M. T. Walter (Orientadora) — CIC/UnB
Prof.ª Dr.ª Aleteia Patricia Favacho de Araujo — CIC/UnB
MsC. Hugo Wruck Schneider — CIC/UnB

CIP — Catalogação Internacional na Publicação

dos Santos, Daniel Dantas Nascimento.

Extração de características de RNAs não-codificadores longos utilizando o algoritmo Random Forest / Daniel Dantas Nascimento dos Santos. Brasília : UnB, 2016.
231 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2016.

1. RNAs não-codificadores longos, 2. RNAs não-codificadores,
3. Aprendizagem de Máquina, 4. *Random Forest*

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Extração de características de RNAs não-codificadores longos utilizando o algoritmo Random Forest

Daniel Dantas Nascimento dos Santos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Engenharia de Computação

Prof.ª Dr.ª Maria Emilia M. T. Walter (Orientadora)
CIC/UnB

Prof.ª Dr.ª Aleteia Patricia Favacho de Araujo
CIC/UnB

MsC. Hugo Wruck Schneider
CIC/UnB

Prof. Dr. Ricardo Pezzuol Jacobi
Coordenador do Bacharelado em Engenharia de Computação

Brasília, 8 de Dezembro de 2016

Dedicatória

Dedico esse trabalho, primeiramente à meus pais, que sempre proveram todas as ferramentas necessárias para meus estudos e que me acompanharam ao longo desta jornada. À minha esposa que sempre foi compreensiva e me apoiou em todos os momentos ao longo deste trabalho. À professora Maria Emilia que além de acreditar no meu potencial teve paciência e dedicação ao me orientar. Aos meus amigos, principalmente meus colegas de faculdade e de intercâmbio, que acompanharam de perto a minha vida acadêmica sempre me ajudando e compartilhando momentos inesquecíveis.

"The most beautiful experience we can have is the mysterious. It is the fundamental emotion that stands at the cradle of true art and true science". Albert Einstein

Agradecimentos

Agradeço primeiramente à professora Maria Emilia por se dispor a me orientar neste projeto e por me apresentar à área multidisciplinar da bioinformática. À meus pais por sempre me incentivarem nos estudos. Ao mestrando Lucas Maciel e doutorando Hugo Schneider por todo o auxílio prestado ao longo deste trabalho. À todos os amigos e familiares que participaram dessa minha jornada acadêmica. Por fim, agradeço à todos os professores e pessoas que de certa forma me ajudaram a chegar até aqui, por sua amizade e apoio no meu período acadêmico.

Resumo

RNAs não-codificantes longos (lncRNAs) são uma classe grande e diversificada de moléculas de RNAs não-codificadores (ncRNAs) com um comprimento de mais de 200 nucleotídeos. LncRNAs tem pouca capacidade de codificar proteínas. Muitos estudos confirmam que o genoma humano contém milhares de lncRNAs que estão envolvidos na regulação de genes e em diversos outros fenômenos nos mecanismos celulares. A identificação e classificação de ncRNAs não é simples, não sendo ainda conhecidas características determinantes para identificar e classificar ncRNAs. Com o advento das tecnologias de sequenciamento avançadas, grande quantidade de sequências não foram ainda analisadas. Neste trabalho, avaliamos características que podem ser utilizadas em métodos de aprendizagem de máquina para predizer lncRNAs. Em particular, usamos o *Random Forest* por ser um dos algoritmos de aprendizagem de máquina mais precisos disponíveis. Além disso, fornece estimativas de quais variáveis são importantes na classificação. Foi desenvolvido um estudo de caso para calcular a performance do modelo proposto para o *Homo sapiens* (humano). Neste trabalho, além de mostrar que o *Random Forest* é um algoritmo apropriado para construção de modelos preditivos, apresentando boa acurácia ao predizer transcritos de lncRNAs e PCTs corretamente, características que podem ser importantes para a classificação dos lncRNAs foram identificadas.

Palavras-chave: RNAs não-codificadores longos, RNAs não-codificadores, Aprendizagem de Máquina, *Random Forest*

Abstract

Long non-coding RNAs (lncRNAs) are a large and diverse class of ncRNA molecules with a length of more than 200 nucleotides. LncRNAs have little ability to encode proteins. Many studies confirm that the human genome contains thousands of lncRNAs that are involved in the regulation of genes and in several other cellular mechanical phenomena. The identification and classification of ncRNAs is not simple, and determinant characteristics to identify and classify ncRNAs are not yet known. With the advent of high-through sequencing technologies, a large number of sequences were not yet analyzed. This research evaluates features that can be used in machine learning methods to predict lncRNAs. In particular, *Random Forest* was used as it provides one of the most accurate machine learning algorithms available. Moreover, it estimates of which variables are important to classification. A case study was developed to measure the performance of the proposed model for the *Homo sapiens* (human). In this work, besides showing that *Random Forest* is an appropriate algorithm for constructing predictive models while accurately predicting both lncRNAs and PCTs transcripts, characteristics that may be important for the classification of the lncRNAs were identified.

Keywords: long non-coding RNAs, non-coding RNAs, Machine Learning, *Random Forest*

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Grandes avanços ocorreram na Biologia Molecular desde a descoberta da estrutura espacial dupla hélice da molécula de DNA por Watson e Crick em 1953 [87]. Na década de 1990 com a criação do projeto Genoma Humano [41], estudos foram realizados para produzir um mapa físico completo de todos os cromossomos e de toda a sequência de DNA dos seres humanos. Esses estudos servem como base para os atuais projetos de sequenciamento de genoma humano, os quais possibilitam ampliar o conhecimento de funções e estruturas de diversas moléculas dos organismos.

O genoma humano contém mais de três bilhões de pares de bases de DNA e toda a informação genética necessária formar seres humanos. O genoma humano foi o primeiro a ser mapeado e sequenciado ao longo de um período de 13 anos de 1990 a 2003. O Projeto Genoma Humano (HGP) [41] foi uma iniciativa internacional inovadora, considerado um dos mais ambiciosos projetos científicos realizado no século passado.

Um genoma contém a informação genética necessário para fazer um organismo vivo, escrito no DNA em código formado de quatro bases ou nucleotídeos. O sequenciamento do genoma de um organismo nos dá uma visão abrangente de sua informação, com a qual podemos melhor compreender a sua evolução, desenvolvimento e funções biológicas. O sequenciamento do genoma humano ajudou os pesquisadores a identificar genes e sequências genéticas importantes, para melhor compreender o seus papeis em doenças, e para investigar as nossas origens usando variações na sequência do DNA.

O Projeto Genoma Humano [85] foi um esforço multinacional com o objetivo de produzir um mapa físico completo de todos os cromossomos humanos, bem como toda a sequência de DNA de humanos. Genomas de outros organismos, tais como bactérias e leveduras, foram estudadas inicialmente, e permitiram aprimorar técnicas laboratoriais e de computação, posteriormente usadas para o genoma humano.

The Genomes OnLine Database (GOLD) [61] monitora de forma centralizada, projetos de genoma e metagenomas em todo o mundo. Ambos, projetos completos e projetos em curso, juntamente com seus metadados associados, podem ser acessados no GOLD por meio de tabelas pré-computadas e uma página de pesquisa. Em setembro de 2009, o GOLD continha informações para mais de 5800 projetos de sequenciamento de genomas, dos quais 1.100 foram concluídos e os seus dados de sequências depositadas em um repositório público. O GOLD continua a se expandir, movendo-se em direção ao objetivo de proporcionar o repositório mais completo de informações sobre sequenciamento genômico.

Em 14 de outubro de 1997, foi lançado pela FAPESP o que viria a ser o o maior projeto científico já realizado no Brasil [29], o sequenciamento genético da bactéria Xylella fastidiosa. Esse projeto contou com o apoio do Fundo Paulista de Defesa da Citricultura (Fundecitrus), e um investimento de US$15 milhões. O Genoma Xylella foi o primeiro sequenciamento de um fitopatógeno (organismo causador de uma doença em uma planta de importância econômica) e ganhou visibilidade internacional [29].

A bactéria gram-negativa Xylella fastidiosa é o principal problema no cultivo de laranja no Brasil por causar a doença clorose variegada dos citros (CVC), conhecida popularmente como a doença do amarelinho. O projeto "Genoma Xylella fastidiosa"foi idealizado devido a importância do cultivo de laranja no Brasil, onde foi proposto o sequenciamento total do genoma deste fitopatógeno bem como o treinamento de mão de obra capacitada na utilização das modernas técnicas de biologia molecular [53].

A Chromobacterium violaceum, é conhecida por sua capacidade de produzir plásticos biodegradáveis reduzir impactos da poluição unir partículas de ouro em áreas de mineração além de combater doenças como o Mal de Chagas e a leishmaniose [20].

Semelhante a iniciativa do sequenciamento da bactéria Xylella fastidiosa foi realizado o sequenciamento do DNA da bactéria Chromobacterium violaceum, microrganismo de grande potencial medicinal, ecológico e industrial. Contou com a criação do Projeto Genoma Brasileiro em 2000 e iniciativas do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq/MCT) com investimentos de R$ 10 milhões. O projeto foi concluido em dezembro de 2001 [20].

O projeto "Genoma funcional diferencial do P. brasiliensis"teve como objetivo geral o mapeamento do genoma funcional e diferencial entre as formas de micélio e levedura de Paracoccidioides brasiliensis [30].

O Paracoccidioides brasiliensis é um fungo de solo que sofre uma alteração dimórfica após a inalação de acolhimento, devido ao aumento da temperatura. Esse fungo é o causador da paracoccidioidomicose (PCM), uma das micoses endêmicas mais importantes da América Latina [30].

A Biologia Molecular é uma área que tem por objetivo estudar a estrutura e funções de proteínas e ácidos nucléicos [21]. Proteínas são moléculas constituídas por uma ou mais cadeias de aminoácidos e realizam funções de transporte de nutrientes, aceleração de reações químicas (enzimas), e construção de estruturas nas células. Os ácidos nucléicos têm a função principal de armazenar informação necessária, prover mecanismos para a criação de proteínas, e também de possibilitar a transferência desta informação para outros organismos, utilizando processos de reprodução celular. Existem dois tipos de ácidos nucléicos, sendo eles: o DNA (ácido desoxirribonucléico) e o RNA (ácido ribonucléico) [71]. Fazem parte do grupo de RNAs os RNAs envolvidos na síntese de proteínas e os que não são traduzidos em proteínas como os ncRNAS (não-codificadores de proteínas).

Com o aprofundamento nos estudos de Watson e Crick, em 1985, Francis Crick propôs o Dogma Central da Biologia Molecular [**?**], que demonstra que através da transcrição, determinadas áreas da molécula de DNA, transformam-se em mRNA (RNA mensageiro) e este por sua vez é sintetizado em proteína através dos RNAs ribossomal (rRNA) e transportador (tRNA), pelo processo conhecido como tradução.

Pesquisas mostram que, no genoma humano, menos de 2% do material genético é transcrito em RNAs codificadores de proteínas, sendo que uma significativa parcela do material genético é transcrito em diversos tipos de ncRNAs [79], e várias classes dife-

rentes de RNA de regulamentação com funções importantes estão sendo descobertas. As regiões não codificadoras de proteínas são denominadas ncRNAs que atualmente são uma importante vertente da biologia molecular, mas que na década de 1980 eram considerados como RNAS lixo (*junk* RNA) não sendo considerados para análise do genoma. RNAs não-codificadores (NcRNAs), mesmo sem traduzirem proteína, possuem papéis importantes nos mecanismos celulares, apresentam uma formação espacial específica que lhes permitam desempenhar papéis reguladores numa grande variedade de reações e processos biológicos, ou por exemplo, a iniciação da tradução, o controle do nível de RNAm, manutenção de células-tronco, cérebro em desenvolvimento, regulação do metabolismo, o apoio à proteína transportes e edição de nucleotídeos.

Compreender o significado deste mundo de RNA não-codificadores é um dos desafios mais importantes da Biologia Molecular hoje em dia. A identificação e classificação de ncRNAs não é tão simples. Os métodos biológicos e computacionais ainda não são capazes de identificar e classificar facilmente ncRNAs, o que afeta diretamente a anotação destes transcritos. RNAs que apresentam sequências de nucleotídeos muito diferentes (sequências primárias), mas semelhantes conformações espaciais (estrutura secundária) executam as mesmas funções celulares. Portanto, ncRNAs precisam ser caracterizados pelas suas estruturas secundárias e não somente pelas suas sequências primárias. Neste contexto, os biólogos utilizam ferramentas diferentes, juntamente com a seu conhecimento para anotar as sequências que parecem ser ncRNAs.

RNAs não-codificantes longos (lncRNAs) são uma classe grande e diversificada de moléculas de ncRNA com um comprimento de mais de 200 nucleótidos que não codificam proteínas. LncRNAs abrangem cerca de 30.000 transcritos diferentes em humanos, por conseguinte, transcritos de lncRNA representam a maior parte do transcritoma de não codificação. lncRNAs podem ser classificados em diferentes subtipos de acordo com a posição e direção da transcrição em relação a outros genes.

LncRNAs estão envolvidos na regulação de genes através de uma variedade de mecanismos. O processo de transcrição do próprio lncRNA pode ser um marcador de transcrição e o lncRNA resultante pode funcionar na regulação da transcrição ou na modificação da cromatina (normalmente através de interações com o DNA e proteínas). LncRNAs podem ligar-se ao RNA complementar e afetar processamento, o *turnover* ou localização do mesmo. A interação de lncRNAs com proteínas pode afetar a função das proteínas e suas localizações, assim como facilitar a formação de complexos de RNA.

LncRNAs podem regular a expressão do gene e a síntese de proteínas em um número de maneiras diferentes. Alguns lncRNAs são altamente expressos, e parecem funcionar como suportes para domínios subnucleares especializados. LncRNAs possuem estruturas secundárias que facilitam as suas interações com o DNA, RNA e proteínas. Um lncRNA também pode ligar-se ao DNA ou RNA de uma maneira específica da sequência.

Devido a importância dos lncRNAs nos organismos celulares e a ausência de características de suas estruturas primárias (sequências de nucleotídeos), existe a necessidade da construção de métodos de identificação de características importantes dos lncRNAs para a criação de métodos computacionais e laboratoriais para sua predição.

1.1 Motivação

LncRNAs são reguladores importantes da expressão dos genes, e têm uma ampla gama de funções em processos celulares e de desenvolvimento, por isso existe a necessidade de criar métodos computacionais e laboratoriais para sua predição. Porém, ainda não se tem clareza sobre os papéis biológicos exercidos pelos lncRNAs, poucos lncRNAs foram caracterizados com detalhes. Assim, ainda são grandes desafios para predizer, identificar e classificar ncRNAs, usando métodos computacionais.

1.2 Problema

Não são conhecidas características das estruturas primárias (sequências de nucleotídeos) de lncRNAs.

1.3 Objetivos

1.3.1 Objetivo Principal

Criar um método de extração de características para lncRNAs baseado em aprendizagem de máquina.

1.3.2 Objetivos Específicos

- Propor e implementar um método de extração de características dos lncRNAs utilizando o algoritmo *Random Forest*;
- Propor métodos de aprendizado de máquina (SVM e *Random Forest*) com as características obtidas do passo anterior.;
- Realizar estudo de caso para lncRNAs em humanos, com o método acima;
- Analisar os resultados obtidos do estudo de caso;
- Comparar os resultados com outros métodos conhecidos na literatura.

1.4 Descrição dos Capítulos

No Capítulo 2, inicialmente serão apresentados conceitos básicos de Biologia Molecular e de Bioinformática. Em seguida, são descritos RNAs não-codificadores, suas classificações, funções e métodos de classificação computacionais, além de bancos de dados que contêm dados de ncRNAs.

No Capítulo 3, são apresentadas noções básicas de Aprendizagem de Máquina e seus quatro paradigmas de aprendizagem. Em seguida alguns métodos de classificação por aprendizagem de máquina serão mostrados. Por fim descrevemos o método *Random Forest* e o SVM, que serão usados neste projeto.

No Capítulo 4, será proposto um modelo de classificação e identificação de características imporatantes para lncRNAs, baseado no *Random Forest*, usando características obtidas na literatura.

No Capítulo 5, serão utilizados dados de humanos para treinar e testar os métodos SVM e *Random Forest* propostos para avaliar o uso das características identificadas no capítulo anterior.

Finalmente, no Capítulo 6, este trabalho será concluído e serão apresentados os trabalhos futuros.

Capítulo 2

RNAs não-codificadores

Neste capítulo conceitos básicos sobre Biologia Molecular serão apresentados, em particular sobre RNAs não-codificadores (ncRNAs) e RNAs não-codificadores longos (lncRNAs). Na Seção 2.1, serão descritos os ácidos nucleicos (DNA e RNA), proteínas e o Dogma Central da Biologia Molecular. Na Seção 2.2, serão mostrados os ncRNAs e suas diferentes classes, tendo como foco os lncRNAs. Por fim, na Seção 2.3, serão apresentadas algumas ferramentas computacionais e banco de dados utilizadas para anotação de ncRNAs.

2.1 Biologia Molecular

Biologia Molecular é uma área que tem como objetivo estudar as estruturas e funções de proteínas e ácidos nucléicos [21]. Esse estudo abrange as reações químicas envolvidas na duplicação do material genético e a síntese de proteinas.

2.1.1 Ácidos nucleicos

Os ácidos nucléicos são polímeros formados a partir de moléculas mais simples, chamadas de nucleotídeos. Um nucleotídeo possui em sua composição uma molécula de açúcar com cinco átomos de carbono (pentose), ligada a um grupo fosfato e uma base nitrogenada [45] Figura 2.1.

O carbono 3' de um nucleotídeo liga-se a um grupo fosfato, que se liga ao carbono 5' de um próximo nucleotídeo, formando assim uma cadeia como pode ser observada na Figura 2.2.

Os ácidos nucléicos têm a função principal de armazenar informação necessária, prover mecanismos para a criação de proteínas, e também de possibilitar a transferência desta informação para outros organismos, utilizando processos de reprodução celular. Existem dois tipos de ácidos nucléicos, sendo eles: o DNA (ácido desoxirribonucléico) e o RNA (ácido ribonucléico) [71]. A Figura 2.3 mostra a diferença entre a pentose encontrada no DNA (desoxirribose) e a pentose ligada ao RNA (ribose), que consiste na presença ou ausência de uma hidroxila (OH) no carbono 2'.

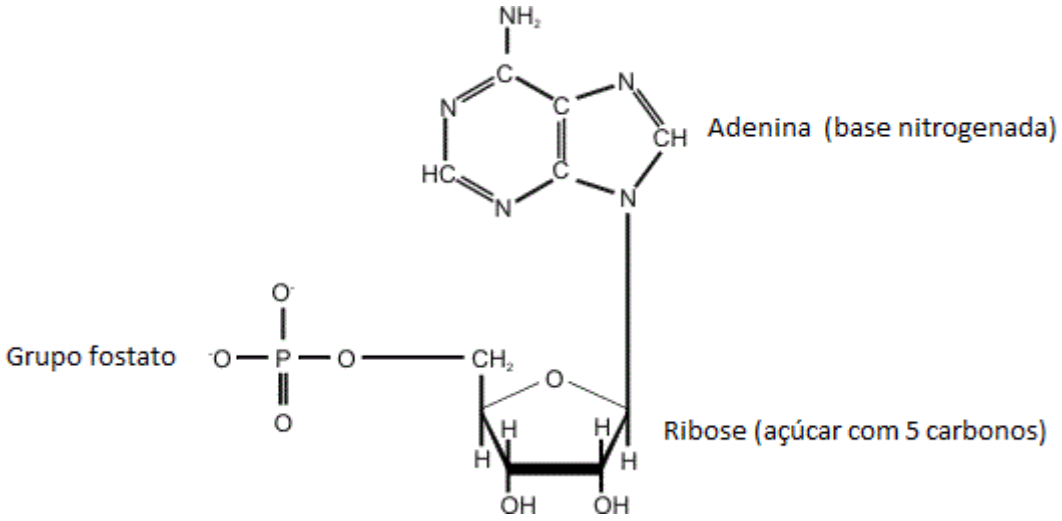


Figura 2.1: Estrutura de um nucleotídeo (Adenina) [33].

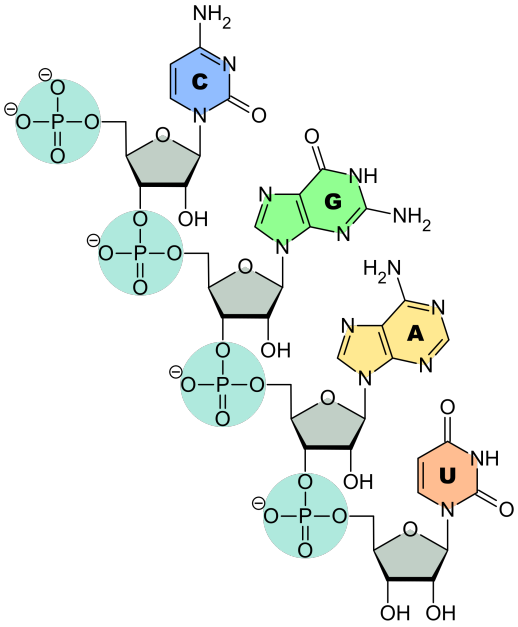


Figura 2.2: Cadeia de nucleotídeos formada pela ligação dos grupos fosfatos [88].

DNA

O DNA é o responsável pelo armazenamento das características genéticas dos seres vivos, além de armazenar as informações necessárias para formar RNAs e proteínas. Como dito antes, o DNA em sua composição possui como sua molécula de açúcar a desoxirribose, a informação no DNA é armazenada como um código composto de suas bases nitrogenadas que são: adenina (A), guanina (G), citosina (C) e timina (T). A pentose do DNA formada por 5 átomos de carbono (1' a 5') onde o carbono 2' liga-se a um átomo de higrogênio. As bases nitrogenadas do DNA emparelham-se aos pares, adenina com timina e citosina com guanina, para formarem pares de bases. Os nucleotideos unem-se por meio de ligações dos grupos fosfatos e estão dispostos em uma longa cadeia devido ao fato do carbono 3'

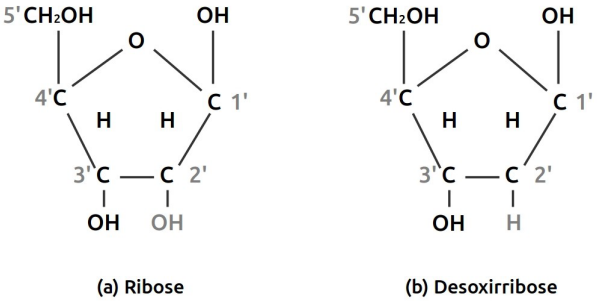


Figura 2.3: Diferença entre moléculas de açúcar com cinco átomos de carbono (pentose), (a) Ribose e (b) Desoxirribose [62].

do primeiro nucleotídeo ligar-se a um grupo fosfato, que se liga ao carbono 5' do próximo nucleotídeo. Devido as ligações das bases nitrogenadas entre duas fitas diferentes, onde Adenina liga-se a Timina e Citosina liga-se a Guanina, a estrutura do DNA é composta de duas longas cadeias de formato helicoidal chamado de dupla hélice [87], veja a Figura 2.4. A estrutura de dupla hélice pode ser comparada com a estrutura de uma escada, com os pares de bases formando os degraus e a ligação entre a desoxirribose e o grupo fosfato formando a estrutura lateral vertical da escada. As regiões que contêm informações necessárias para codificar proteínas são chamadas de genes.

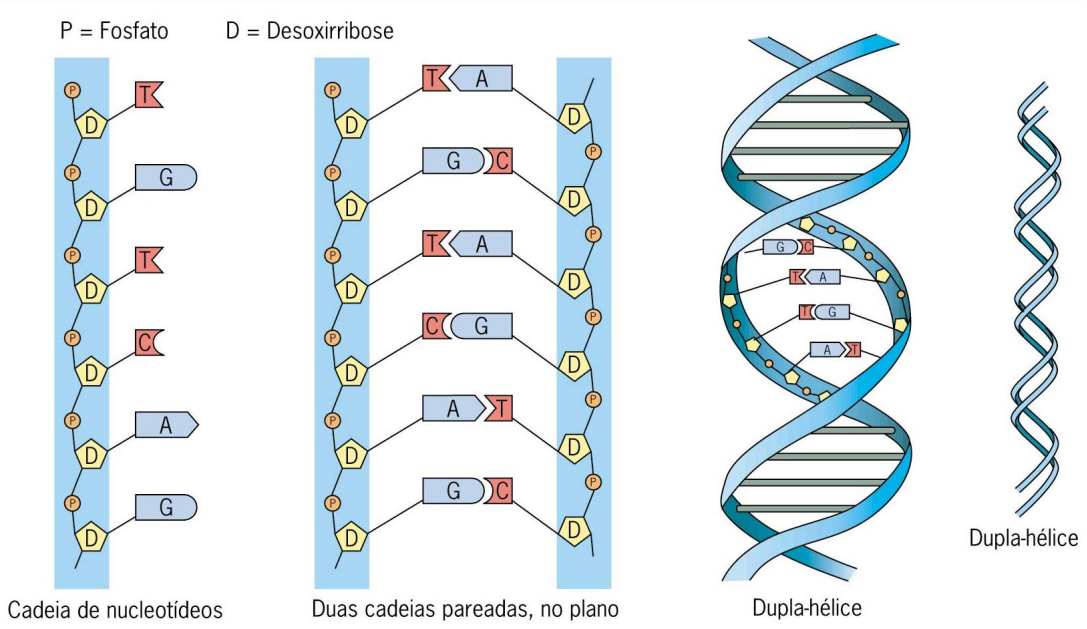


Figura 2.4: Estrutura do DNA num plano e sua estrutura dupla hélice [6].

Genes

Gene, no contexto do processo de síntese de proteínas, corresponde a uma região do DNA que pode ser transcrito em um pré-mRNA. As regiões do DNA situadas entre os genes são chamadas de regiões intergênicas. Como dito antes, nem toda a informação de

um gene é utilizada para a produção de proteínas e parte do pré-mRNA é descartado no processo de *splicing*. Com base nessa informação, os genes contêm partes denominadas éxons e outras denominadas íntrons. Um éxon é um trecho contíguo de uma sequência de DNA que vai ser utilizado na síntese do mRNA. Um íntron é um trecho do DNA que é descartado no processo de *splicing*. De acordo com a posição onde se encontram dentro do gene, os éxons podem ser classificados em quatro classes: éxon inicial (primeiro éxon do gene), éxon final (último éxon do gene), éxon interno (qualquer éxon situado entre os éxons inicial e final) e éxon único (éxon componente de um gene constituído por um único éxon). As regiões correspondentes aos éxons de uma sequência de DNA são chamadas de regiões codificadoras.

Existem outras porções de sequências de DNA com papéis variados na expressão gênica, além dos éxons e dos introns Figura 2.5. Essas regiões são conhecidas como regiões funcionais. Os genes podem codificar mais de uma proteína devido ao processo chamado *splicing* alternativo, onde vários mRNAs maduros (mRNAs obtidos após *splicing* dos íntrons) diferentes podem ser sintetizados a partir de um mesmo gene, utilizando subconjuntos distintos do conjunto original de éxons. A seguir é apresentado a relação de algumas delas:

- Promotor: localiza-se no início de um gene. A enzima RNA-polimerase liga-se a esta região para dar início à transcrição;
- Terminador: localiza-se no final de um gene e sinaliza o final do processo de transcrição.

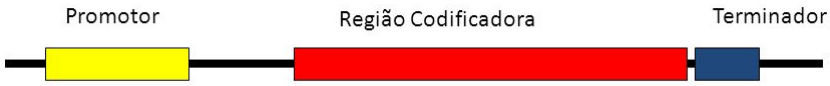


Figura 2.5: Estrutura do gene [74].

RNA

O RNA é uma parte importante da síntese de proteínas da célula. O RNA difere do DNA pelo fato da molécula de RNA ser formada por uma cadeia única de nucleotídeos além de possuir em sua composição, como sua molécula de açúcar, a ribose. A informação no RNA é armazenada como um código composto pelas mesmas três bases nitrogenadas do DNA, adenina (A), guanina (G), citosina (C), porém apresenta a uracila (U) em vez da Timina (T) [46] (Figura 2.6).

Diferentemente do DNA, encontramos vários tipos de moléculas de RNA, cada qual executando uma função diferente [50]. Fazem parte do grupo dos RNAs, aqueles envolvidos na síntese protéica, como é o caso do RNA mensageiro (mRNA), o RNA ribossomal (rRNA) e o RNA transportador (tRNA), além dos que não são traduzidos em proteínas como os ncRNAS (não-codificadores de proteínas). Os RNAs que participam da síntese de proteínas possuem diversas funções em um organismo tais como a constituição do ribossomo (rRNA), o transporte de aminoácidos utilizados na síntese de proteínas (tRNA), o transporte de informações codificadas pelo DNA para a síntese protéica (mRNA), além de diversos papéis em processos de regulação gênica [71].

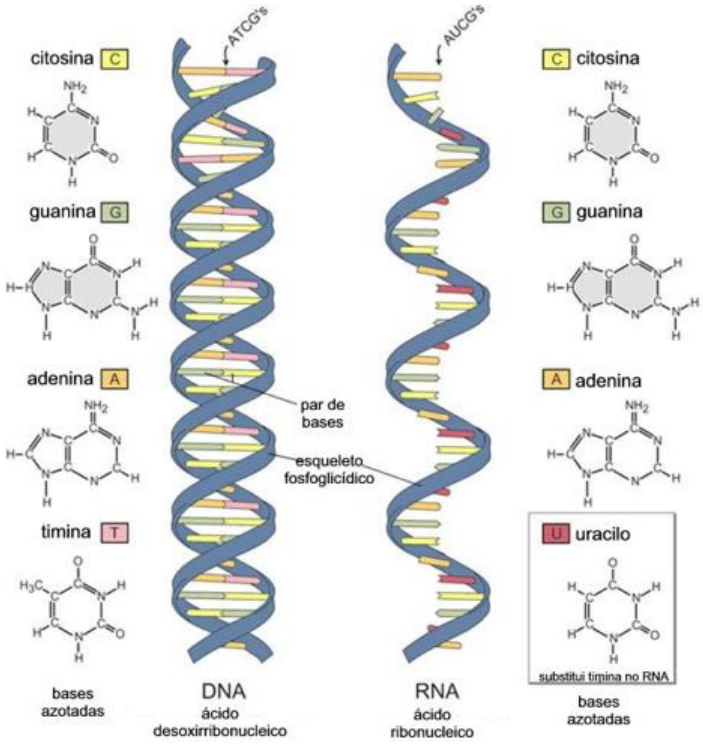


Figura 2.6: Estrutura do DNA e RNA [75].

2.1.2 Proteínas

Proteínas são macromoléculas constituídas por uma ou mais cadeias de aminoácidos e realizam funções de transporte de nutrientes, aceleração de reações químicas (enzimas), eliminação resíduos tóxicos e construção de estruturas nas células [71]. Todo aminoácido é formado por um átomo de carbono central (carbono alfa), que possui anexado ao mesmo um átomo de higrogênio (H), um grupo amina (NH2), um grupo carboxila (COOH) e a uma cadeia lateral, sendo esta responsável por diferenciar um aminoácido do outro (Figura 2.7). As proteínas são formadas a partir de 20 tipos de aminoácidos (Figura 2.8).

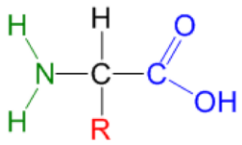


Figura 2.7: Estrutura do aminoácido formado por: um átomo de carbono central (carbono alfa), um átomo de higrogênio (H), um grupo amina (NH2), um grupo carboxila (COOH) e um radical R [14].

2.1.3 Dogma Central da Biologia Molecular

Em 1985, Francis Crick propôs o Dogma Central da Biologia Molecular [87], o qual explica como ocorre o fluxo de informações genéticas. Esse estudo propõe o processo de duplicação de uma molécula de DNA (replicação); o processo de transcrição, onde ocorre

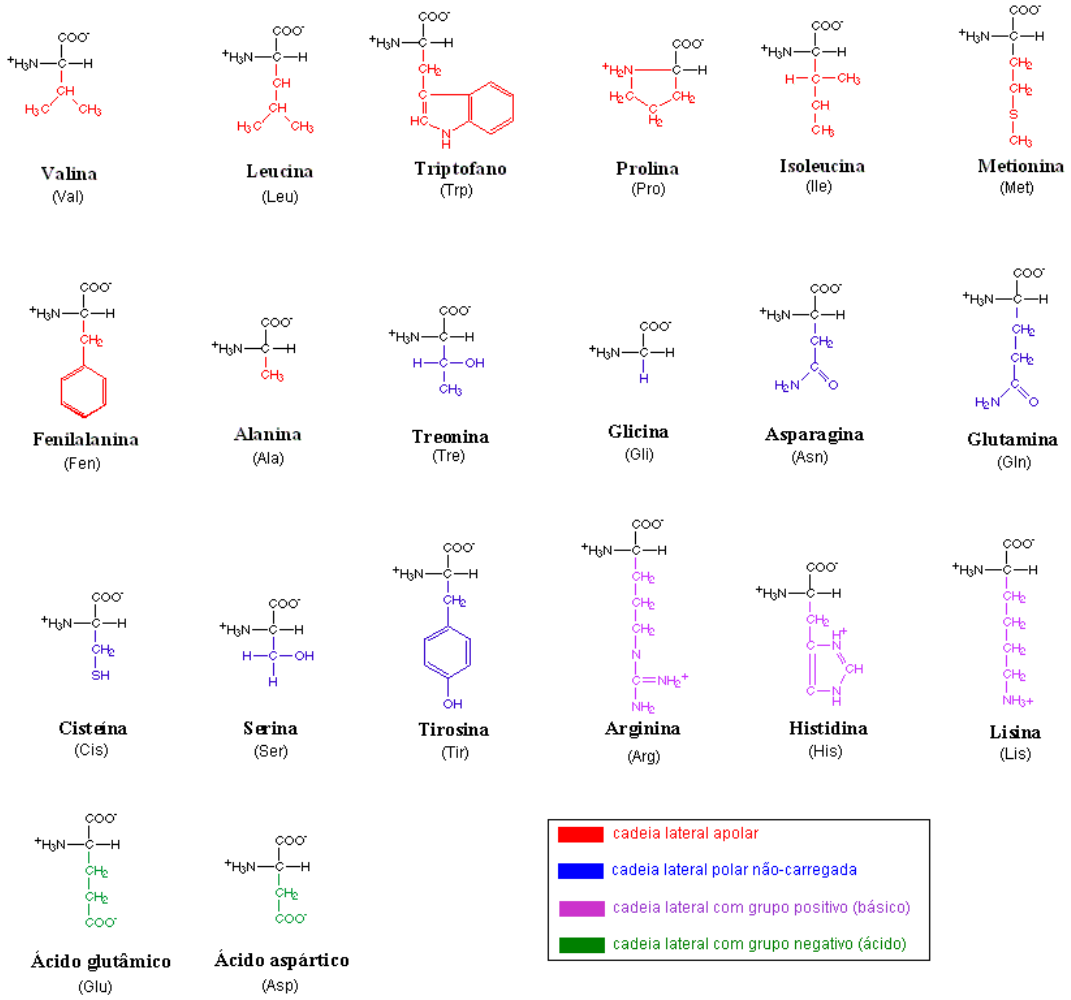


Figura 2.8: Os 20 tipos de Aminoácidos que formam Proteínas [37].

a produção de RNA; como também, o processo de tradução, no qual temos a produção de proteínas a partir de RNAs (tRNA e rRNA) (Figura 2.9).

No processo de replicação, ocorre a quebra da estrutura dupla hélice da molécula de DNA, o que é feito por uma enzima chamada helicase, que quebra as ligações de hidrogénio que mantêm as bases nitrogenadas complementares do DNA (A ligado a T, C ligado a G). Esse processo faz com que uma abertura em formato de 'Y' seja criada. A enzima chamada DNA primase liga-se às cadeias de DNA para inicia a síntese que adiciona os primeiros nucleotídeos as fitas de DNA e em seguida a enzima chamada DNA polimerase fica liga-se a nucleotídeos dispersos no núcleo as fitas de DNA. Ao final do processo, são produzidas duas moléculas idênticas, cada dupla fita de DNA nova formada será metade antiga e metade nova. Devido a esse fato, o processo de replicação é considerado semi-conservativo (Figura 2.10).

Na transcrição, diferentemente da replicação do DNA, um RNAm (RNA mensageiro)

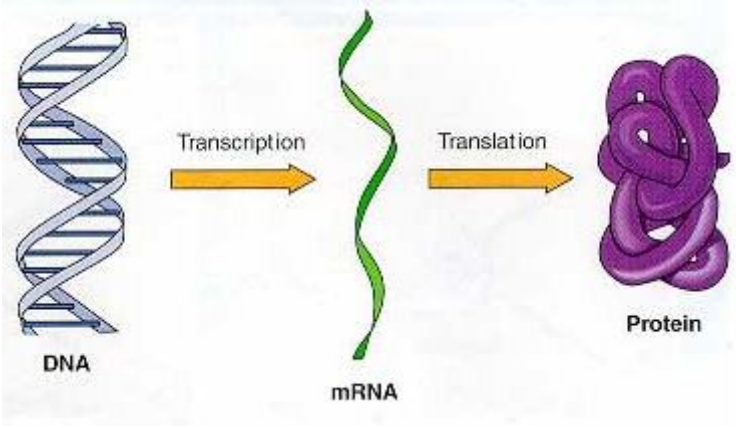


Figura 2.9: Processos de Tradução e Transcrição [59].

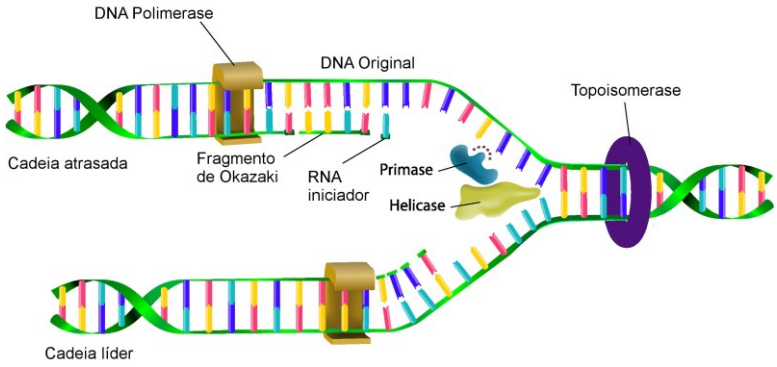


Figura 2.10: Processo da Replicação do DNA [38].

é transcrito a partir de uma das cadeias da molécula de DNA. O RNA é denominado RNA mensageiro porque ele carrega a informação genética do DNA para os ribossomos, onde a informação é utilizada para produzir proteínas. A transcrição é iniciada quando a enzima RNA polimerase liga-se a região de um gene do DNA a qual é chamada de promotor, que normalmente é precedida por uma sequência de TA (chamada de TATA box) [21]. A enzima RNA polimerase então gera uma molécula de RNAm por meio de uma sequência complementar de bases nitrogenadas. Este processo ocorre devido à RNA polimerase ler a cadeia de DNA desenrolada e constrói a molécula de mRNA por meio da adição de nucleotídeos à sua cadeia, utilizando-se de pares de bases complementares. O fim da transcrição ocorre quando RNA polimerase atravessa uma região terminal o que nada mais é do que uma sequência de terminalização no gene. Neste momento, nenhuma outra base nitrogenada é incorporada ao RNA e a cadeia de RNAm está completa. A molécula de RNA é então liberada e imediatamente a molécula de DNA volta a se enrolar por completo (Figura 2.11).

Durante a tradução, ocorre a leitura da informação (sequência de nucleótidos) passada do DNA como mRNA para a síntese de proteína, onde a mensagem será traduzida em uma série de aminoácidos unidos por ligações peptídicas, daí a origem do nome. Cada grupo de três bases em um RNAm constitui um códon, e cada códon especifica um aminoácido particular. A sequência de RNAm é assim utilizado como um molde para montar a cadeia

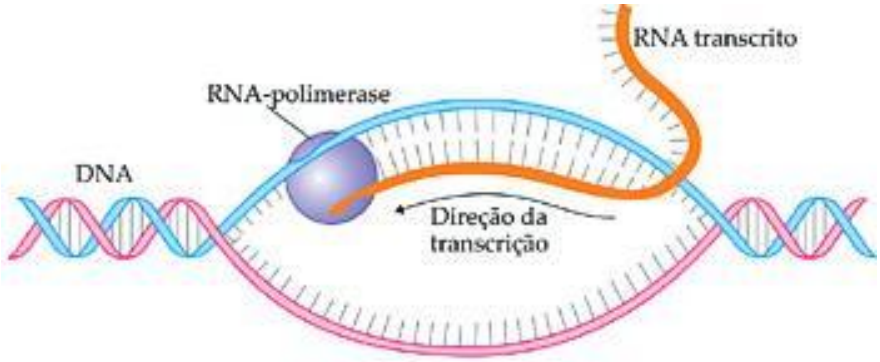


Figura 2.11: Processo de Transcrição de uma Molécula de DNA em uma Molécula de RNAm [12].

de aminoácidos que formam uma proteína. A síntese do mRNA ligado a tRNAs ocorre nos ribossomos, que são complexos citoplasmáticos constituídos de RNAs ribossomais (rRNAs) e proteínas. No processo de tradução, primeiramente o mRNA liga-se entre as duas subunidades do ribossomo, onde cada códon do mRNA é pareado com o anticódon correspondente que está presente em moléculas de tRNA [73]. Os ribossomos funcionam como uma linha de montagem de uma fábrica, usando como entradas o mRNA e o tRNA e como saída uma cadeia linear de uma proteína [71]. O ribossomo se move ao longo do RNAm, combinando 3 pares de bases de cada vez e adicionando os aminoácidos à cadeia polipeptídica. Esse processo é interrompido quando o ribossoma atinge um dos códons finalizadores (UGA, UAA ou UAG), com isso o polipeptídeo e o mRNA são liberados deixando o ribossomo disponível para uma nova síntese protéica (Figura 2.12).

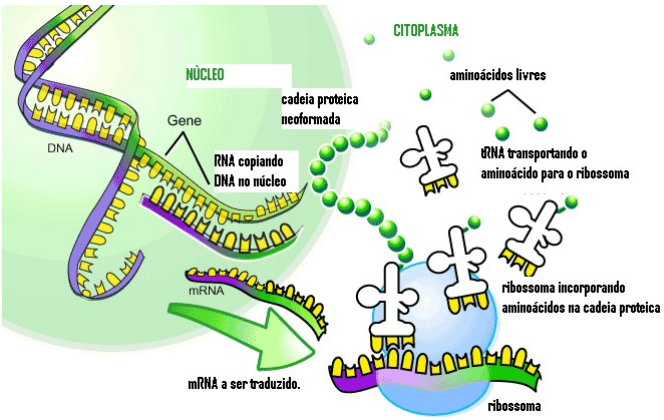


Figura 2.12: Processo de Tradução de uma Molécula de RNAm em uma proteina [15].

2.2 RNAs não-codificadores

NcRNA é qualquer molécula funcional de RNA que não será traduzida em uma proteína, possuindo funções biológicas diversas. Então, os genes de ncRNAs produzem RNAs funcionais em vez de codificar proteínas [22]. Estudos revelaram que cerca de 98% do que

é transcrito pelo genoma humano é constituído de ncRNAs [54]. Os mRNAs são codificadores de proteínas, já os tRNAs e os rRNAs, embora envolvidos no processo de síntese, não codificam proteínas. Diferente do que se pensava nas decadas de 1980 e 1990, quando as regiões não codificadoras (não envolvidos diretamente com a síntese de proteínas) eram chamadas de DNA lixo (*junk* DNA) [71], pesquisas iniciadas nos anos 2000 vêm mostrando que estas regiões não codificadoras desempenham papéis importantíssimos nos organismos [71, 77].

RNAs não-codificadores, mesmo sem traduzirem proteínas possuem papéis importantes nos mecanismos celulares, agem diretamente na célula em funções estruturais, catalíticas ou regulatórias [22, 89], apresentam uma formação espacial específica que lhes permitem desempenhar papéis reguladores numa grande variedade de reações e processos biológicos, por exemplo, a iniciação da tradução, o controle do nível de RNAm, manutenção de células-tronco, cérebro em desenvolvimento, regulação do metabolismo, o apoio à proteína transportes e edição de nucleotídeos.

2.2.1 Classificações de ncRNAs

RNAs não codificadores são moléculas de RNA, que são transcritas, mas não são traduzidas em proteínas. Classes de ncRNAs podem ser distinguidas por suas funções, que dependem diretamente da estrutura e comprimento das suas moléculas, e da composição da sua sequências. Esses ncRNAs podem ser divididos em dois grupos principais; os ncRNAs pequenos ($<$ 200 nucleotídeos) e os ncRNAs longos ($>$ 200 nucleotídeos). Apesar de identificados e a eles serem atribuídos papeis de grande importância, ainda no início dos projetos que envolviam o sequenciamento de genomas inteiros, a caracterização em massa dos RNAs não codificadores foi abandonada por ser complexa, não abundante e principalmente instável.

Nesse contexto, havia pouca motivação para o estudo dessas moléculas [22]. Entretanto, com o passar do tempo, inúmeras descobertas sobre ncRNAs foram feitas, com as mais diversas funções. Atualmente, o número e a diversidade de genes de RNAs que não codificam proteínas são alvos de inúmeras pesquisas. Independente de classificações, a quantidade de ncRNAs identificados cresce rapidamente na literatura. As descobertas mais notáveis envolvendo RNAs estruturais estão relacionadas ao desenvolvimento do sistema nervoso, corroborando a observação de que a quantidade de regiões não-codificadoras é proporcional à complexidade aparente dos organismos [54].

NcRNAs pequenos

Os ncRNAs pequenos mais conhecidos e suas funções:

- RNA transportador (tRNA): São responsáveis pela tradução da informação genética recebida pelo RNAm, traduzindo os códons do RNAm em aminoácidos que serão adicionados a proteína na síntese protéica;
- RNA ribossomal (rRNA): É o componente central do ribossomo. Sua função consiste em prover um mecanismo para decodificar o mRNA em aminoácidos e interagir com os tRNAs durante a tradução. Atuam na catalisação, reconhecimento da síntese protéica, e exercem um papel estrutural;

- *small* nuclear RNA (snRNA): É encontrado no núcleo de uma célula. Eles estão envolvidos no processo de *splicing* do pré-mRNA, em que os íntrons de um transcrito primário são eliminados, resultando no mRNA maduro. A estrutura secundária desses RNAs é altamente conservada nos organismos. Alguns deles, conhecidos como U1, U2, U4, U5 e U6, são essenciais para o *splicing* do pre-mRNA;
- *small* nucleolar RNA (snoRNA): Pequenas moléculas que realizam modificações químicas com o objetivo de promover a maturação de rRNAs, além de outros ncRNAs, tal como o tRNA tornando-os ativos. Acredita-se que eles originam-se dos íntrons do mRNA;
- microRNA (miRNA): Atuam na regulação gênica. São parcialmente complementares a uma ou mais moléculas de mRNA e sua principal função é reduzir a expressão de genes codificantes, inibindo a tradução de mRNAs;
- *small interfering* RNA (siRNA): Atuam na regulação gênica, porém reduz a expressão de genes codificadores degradando o mRNA em vez de inibir sua tradução;
- *piwi-interacting* RNA (piRNA): Pequenas moléculas de RNA existentes basicamente em células dos mamíferos. Atuam na regulação gênica. Mais especificamente, eles atuam no silenciamento de genes capazes de se auto-duplicar no interior do genoma;
- *small non-messenger* RNAs (snmRNAs): São classes de RNAs com funções de regulação;
- *small Cajal body-specific* RNA (scaRNA): Tem função similar à dos snoRNAs. Sua estrutura é formada por ambas as características dos tipos de snoRNAs: C/D box e H/ACA box.

NcRNAs longos

LncRNAs foram considerados como lixo não funcional inicialmente, e agora, a sua presença e importância ainda é debatida [4]. RNAs não-codificantes longos (lncRNAs) são uma classe grande e diversificada de moléculas de ncRNAs com um comprimento de mais de 200 nucleótidos. Os lncRNAs são transcritos que apresentam extremidades tanto de 5' para 3', como ao contrário, podendo sofrer *splicing*. Entretanto, podem apresentar *Open Reading Frame* (ORF) suficiente para codificar proteínas, tendo seu tamanho variando de 200 a 100.000 pares de bases [56].

Atualmente, ainda não se sabe muito a respeito dos papéis exercidos pelos lncRNAs [64], mas sabe-se que muitos transcritos são associados a lncRNAs e possuem um baixo poder de síntese de proteínas [64, 91]. LncRNAs abrangem cerca de 30.000 transcritos diferentes em humanos, por conseguinte, transcritos de lncRNA representam a maior parte dos transcritomas não codificadores. Os lncRNAs podem ser transcritos a partir de regiões distantes dos genes codificadores, dentro dos transcritos ou de genes a partir de íntrons. Eles podem exercer sua ação a partir da região de origem, regulando seus alvos. Já os lncRNAs que são derivados da fita de DNA oposta à de um gene codificador são conhecidos como transcritos antisenso naturais (NATs) e regulam o gene ao qual o lncRNA se sobrepõe [56]. Apesar de serem menos conservados do que genes codificadores em relação à sequência de nucleotídeos, os lncRNAs apresentam uma conservação maior em suas estruturas secundárias [81].

A grande maioria dos lncRNAs que já possui sua função caracterizada estão envolvidos em regulação. Esses lncRNAs são associados a um complexo de remodeladores de cromatina, ou seja, um grupo de genes que reestruturam os nucleossomos de modo a compactar mais ou menos a cromatina, determinando o nível de transcrição gênica de uma região definida do cromossomo. As interações entre proteínas e lncRNAs poderiam resultar em mudanças conformacionais que seriam úteis para distinguir a especificidade da região alvo. De forma resumida, os lncRNAs serviriam como guias para os complexos remodeladores de cromatina, pois esses não possuem capacidade de ligação ao DNA, não reconhecendo suas regiões alvo de forma isolada [56].

Estudos têm mostrado que lncRNAs desempenham papéis reguladores importantes em diversos processos celulares, além de processos como a remodelação da cromatina, participam também da transcrição, processamento pós-transcricional e tráfico intracelular [34, 64]. Portanto, os lncRNAs vêm sendo recentemente considerados como reguladores chave de diversos processos biológicos [90]. LncRNAs podem ser classificados em diferentes subtipos de acordo com a posição e direcção da transcrição em relação a outros genes. Há cinco diferentes categorias de classificação dos lncRNAS [64]:

(a) Senso: quando o lncRNA se sobrepõe a um gene na mesma fita;

(b) Antisenso: quando o lncRNA se sobrepõe a um gene na fita oposta;

(c) Bidirecional: quando o lncRNA e o gene são expressos juntos e estão em fitas opostas;

(d) Intrônico: quando o lncRNA está localizado dentro de uma região intrônica;

(e) Intergênico (*long intergenic* ncRNA - lincRNA): quando o lncRNA situa-se entre dois genes.

2.2.2 Estrutura

As evidências apontam que os ncRNAs, desempenham papéis importantes em várias atividades celulares. Técnicas de sequenciamento de alto desempenho resultaram na geração de grandes quantidades de dados em transcritos. Por conseguinte, é desejável, não só distinguir RNAs que codificam proteína dos que não codificam (ncRNAs), mas também para atribuir RNAs não codificante (ncRNA) a suas respectivas classes (famílias). Embora existam vários algoritmos disponíveis para esta tarefa, sua classificação continua a ser uma grande preocupação. Os ncRNAs tendem a dobrar-se de formas diferentes em suas estruturas secundárias, em parte, porque os RNAs precisam dessa estrutura para serem funcionais, notando-se que essas estruturas são pequenas [49]. O estudo da estrutura de um ncRNAs é de extrema importância para a classificação de qual família ele pertence.

Foram criadas diferentes abstrações da estrutura dos ncRNAs. As três mais usadas são:

- Estrutura primária: a sequência de bases que define a molécula. Essa sequência é gerada pelos sequenciadores automáticos;
- Estrutura secundária: pode-se representá-la em 2D, equivale às ligações entre os pares de bases complementares;

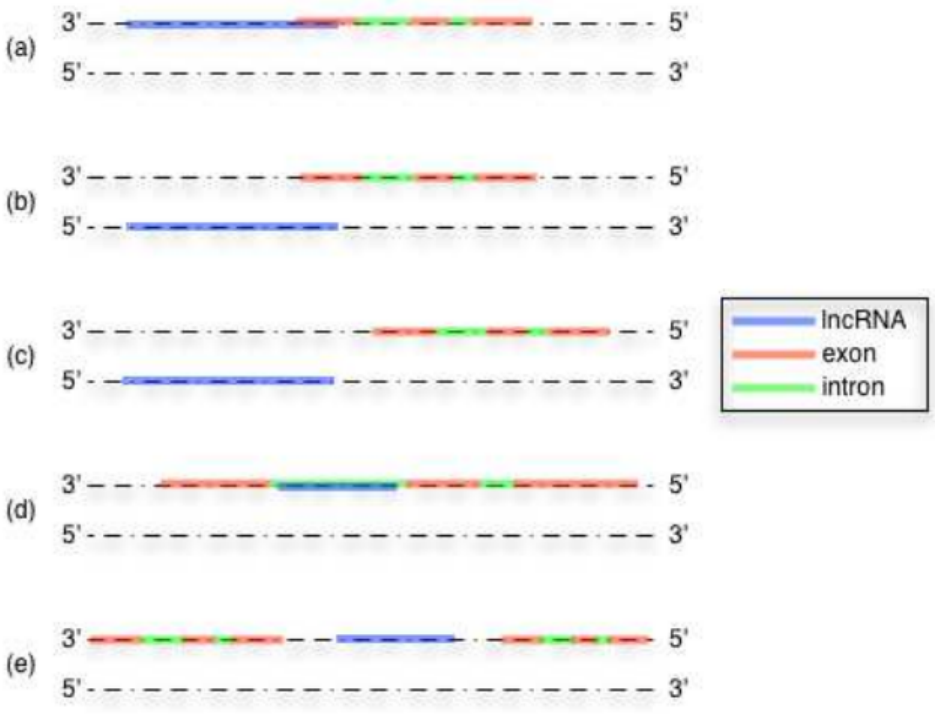


Figura 2.13: Cinco categorias de lncRNA: (a) senso; (b) antisenso; (c) bidirecional; (d) intrônico; e (e) intergênico [69].

- Estrutura terciária: representação espacial 3D de um RNA.

Os componentes estruturais de uma estrutura secundária de RNA são:

1. talo (*stem*): Contém pares de bases complementares [3];
2. alça (*loop*): Local de não pareamento das bases [3];
3. grampo (*hairpin*): Um loop encerrado por uma hélice [3].

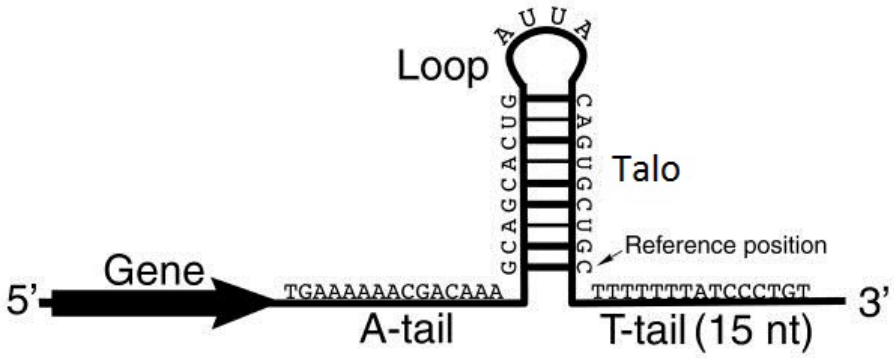


Figura 2.14: Estrutura dos ncRNAs [80].

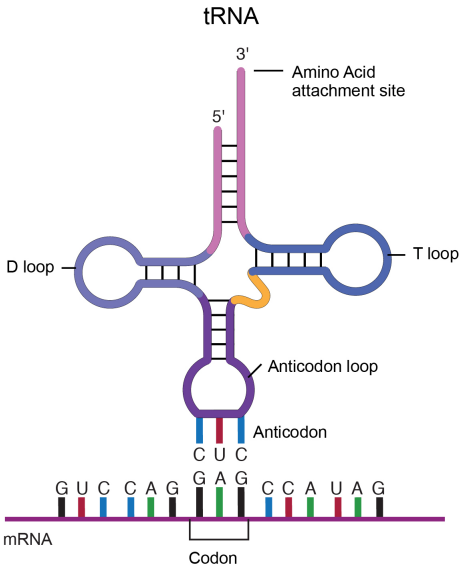


Figura 2.15: Estrutura espacial do RNA Transportador [39].

2.3 Ferramentas computacionais e Banco de Dados para Anotação de ncRNAs

Nesta Seção serão descritas algumas ferramentas computacionais e banco de dados utilizados para a anotação de ncRNAs.

2.3.1 Métodos Computacionais

A presença de ncRNAs nos diversos reinos dos seres vivos é bem documentada, porém a função de cada classe de ncRNAs está longe de ser totalmente conhecida. Métodos computacionais são capazes de contribuir bastante com a caracterização dos ncRNAs, principalmente, com a criação de dados com as tecnologias modernas de sequenciamento. Ferramentas de detecção, análise e integração de dados são de grande importância para aplicações da gama de dados que temos disponíveis atualmente.

O problema da classificação de ncRNA é feito por métodos computacionais que dependem dos dados disponíveis das sequências que estão sendo analisadas. Os ncRNAs podem ser preditos por meio de ferramentas computacionais que buscam características importantes presentes nessa molécula, tais como a presença de promotor, terminador em regiões intergênicas ou em regiões antisenso a ORFs entre outras.

Os métodos computacionais para anotar ncRNAs sofrem de problemas similares aos dos métodos experimentais. A Bioinformática não possui métodos únicos para identificação e classificação de ncRNAs, embora alguns critérios sejam usados, como o fato de que ncRNAs não possuem em geral ORFs longas.

Grande parte dos programas para detecção e análise dos ncRNAs dependem de comparações entre moléculas da mesma família com um certo grau de similaridade. Entretanto, ncRNAs de uma mesma família podem apresentar uma mesma estrutura, mas uma sequência primária (estrutura primária) diferente. Isso faz com que sejam necessárias abordagens que considerem a análise estrutural dos mesmos.

A utilização de ferramentas de alinhamento entre sequências é bastante comum para a identificação de ncRNAs, por exemplo o BLAST que será tratado a seguir. Porém essa abordagem passa a ser limitada, pois o número de ncRNAs já caracterizados é baixo, e existe uma baixa conservação da estrutura primária de várias famílias de ncRNAS.

Os métodos de Bioinformática utilizam uma combinação de diversos métodos computacionais que caracterizem os ncRNAs por meio de diferentes métodos. Depois os biólogos, analisam todas as informações geradas pelos métodos para decidir quais RNAs provavelmente são não-codificadores.

Em seguida, são destricas ferramentas para identificar e classificar ncRNAs.

BLAST

The Basic Local Alignment Search Tool (BLAST) [31] encontra regiões de similaridade local entre sequências. O programa compara as sequências de nucleótidos ou sequências de proteínas a sequências de banco de dados e calcula a significância estatística dos resultados. BLAST pode ser usada para inferir relações funcionais e evolutivas entre sequências, bem como ajudar a identificar os membros de famílias de genes.

BLAST é um dos programas de Bioinformática mais amplamente utilizado para a busca de sequência. Este programa aborda um problema fundamental na pesquisa Bioinformática. O algoritmo de heurística que utiliza é muito mais rápido do que outras abordagens, tais como o cálculo de um alinhamento óptimo. Esta ênfase na velocidade é vital para fazer o algoritmo prático sobre os enormes bancos de dados genômicos actualmente disponíveis, embora algoritmos subsequentes podem ser ainda mais rápidos.

Usando um método de heurística, BLAST encontra sequências semelhantes, localizando partes curtas idênticas entre as duas sequências. Este processo de encontrar sequências similares é chamado de *seeding*. É após esta primeira partida que BLAST começa a fazer alinhamentos locais. Ao tentar encontrar semelhança em sequências, conjuntos de letras comuns, conhecidos como *words*, são muito importantes.

O método do BLAST é dividido em três grandes etapas. Na primeira, são encontrados sequências pequenas de tamanhos fixados (*words*) que ocorrem na sequência de consulta. Na segunda etapa essas palavras são usadas para fazer um busca pela mesma (*query*) em todas as sequências de um banco de dados (*subject*). Em seguida, são feitas extensões, com espaços (*gaps*), em ambos os lados da sequência de consulta em relação à sequência do banco de dados, mantendo um escore minimo. Essas extensões são, então, ligadas, produzindo alinhamentos maiores, porém, ainda mantendo um escore mínimo [3].

Diferentes tipos de BLAST estão disponíveis de acordo com as sequências de consulta. Por exemplo, na sequência da descoberta de um gene previamente desconhecido de um certo animal, um cientista normalmente realiza uma pesquisa BLAST do genoma humano para ver se os seres humanos portam um gene similar. BLAST identifica sequências no genoma humano que se assemelham o gene do animal com base na similaridade de sequência.

Os diferentes tipos de BLAST e suas funções são:

- **blastp**: Utilizado para comparação de sequências de aminoácidos com um banco de dados de proteínas;
- **blastn**: Utilizado para comparação de sequências de nucleotídeos com um banco de dados de nucleotídeos;

- **blastx**: Utilizado para comparação de sequências de nucleotídeos traduzidos em todas as ORFs, com um banco de dados de proteínas;
- **tblastn**: Utilizado para comparação de sequências de proteínas com um banco de dados de sequências de nucleotídeos traduzidos em todas as suas ORFs;
- **tblastx**: Utilizado para comparar as ORFs de sequências de nucleotídeos com todas as ORFs de um banco de dados de nucleotídeos.

Infernal

Infernal (*"INFERence of RNA ALignment"*) [57] é para pesquisar bancos de dados de sequência de DNA, para estrutura de RNA e semelhanças de sequência. É uma implementação de Gramática Estocástica Livres de Contextos (SCFG, *"Stochastic Context-Free"* Grammars) chamados de CMs. O Infernal usa esses Modelos de Covariância (*Covariance Models - CMs*) para criar novos alinhamentos de sequência múltipla baseadas em estrutura ou para procurar as semelhanças entre as estruturas secundárias das famílias de RNAs, de modo que, em muitos casos, é mais capaz de identificar homólogos de RNA que conservam a sua estrutura secundária do que a sua sequência primária.

Ao procurar por RNAs estruturais homólogas em bancos de dados de sequência, é desejável obter tanto a conservação da sequência primária quanto a estrutura secundária. As ferramentas geralmente mais utilizadas que integram sequência e estrutura tomam como entrada qualquer RNA, e constroem automaticamente um sistema de pontuação estatística adequado que permite a classificação quantitativa de homólogos putativos num banco de dados de sequência. As SCFGs fornecem um quadro estatístico para combinação de sequências e informações de conservação de estrutura secundária em um único sistema de pontuação consistente.

Uma utilização do Infernal é de anotar RNAs em genomas usando o banco de dados Rfam [25], que contém centenas de famílias de ncRNAs. o Rfam segue uma estratégia de perfil de sementes, em que um alinhamento bem anotado de "semente"de cada família é avaliado, e um CM construído a partir de que o alinhamento da semente é utilizado para identificar e alinhar membros adicionados da família.

Infernal é composto por vários programas que são combinados, seguindo quatro passos básicos:

- *cmbuild*: Construir um CM a partir de um alinhamento estrutural;
- *cmcalibrate*: Calibra a CM para a pesquisa de homólogos;
- *cmsearch*: Pesquisa bancos de dados para homólogos putativos;
- *cmalign*: Alinhar homólogos putativos para um CM.

O *cmbuild* realiza a construção do CM, no qual um alinhamento múltiplo de RNAs no formato Estocolmo (*Stockholm*) é o dado de entrada, e gera-se, então um arquivo de saída contendo o CM, o qual será usado por outras funções do Infernal. Com o arquivo obtido e um arquivo contendo as sequências a serem analisadas o *cmsearch* realiza a busca em bases de dados por possíveis homólogos. O *cmsearch* busca as sequências que geraram hits com alta pontuação para o CM usado e os alinhamentos para cada hit são retornados. O infernal também possui uma ferramenta chamanda O *Rsearch* que realiza buscas em uma

base de dados de nucleotídeos por RNAs homólogos utilizando tanto a estrutura primária quanto na estrutura secundária [3].

Vienna

O Vienna é um servidor que fornece programas, serviços web e bancos de dados, relacionados com o trabalho em estruturas secundárias de RNAs. Esse pacote tem várias ferramentas, nas quais dobramentos são feitos utilizando um algoritmo de predição basendo na energia livre do RNA, e nas probabilidades de pareamento de bases [3].

Os vários algoritmos oferecidos são geralmente acessados através de diversas linhas de comando, mas o pacote também fornece uma biblioteca em C que pode ser utilizada para o desenvolvimento de novos programas.

É levado em consideração o fato de que as estruturas dos ncRNAs apresentam duas características: a estabilidade termodinâmica e a conservação da estrutura secundária.

O Vienna fornece 3 pacotes: RNAz, RNAfold e o RNAalifold. O pacote RNAz realiza a predição de estrutura baseada na energia mínima livre (*Minimun Free Energy - MFE*). O RNAz é utilizado para detectar estruturas funcionais de RNAs em múltiplos alinhamentos de sequências nucleotídicas. O servidor fornece acesso a um pipeline para análise completo e totalmente automático que permite não apenas analisar alinhamentos únicos em uma variedade de formatos, mas também realizar telas complexas de grandes regiões genômicas.

O pacote RNAfold calcula estruturas secundárias de energias mínimas livres e tem a função de particionar os RNAs fazendo um dobramento bidimensional utilizando-se de um algoritmo de programação dinâmica. O programa lê sequências de RNA, calcula a sua estrutura mínima de energia livre (MFE) e imprime a estrutura MFE na notação de suporte e sua energia livre. O RNAfold oferece várias possibilidades de controlar a estrutura espacial por parte do usuário, como locais da estrutura secundária onde ocorre o pareamento de nucleotídeos para a formação das hélices [36].

Por ultimo, o pacote RNAalifold constrói uma estrutura bidimensional consenso, a partir do alinhamento múltiplo de sequências de RNA. O algoritmo utiliza informações termodinâmicas e filogenéticas para determinar a estrutura da predição. Uma estrutura secundária de consenso é inferida a partir do alinhamento [3].

2.3.2 Banco de Dados

Na literatura, há diversos bancos de dados com informações de ncRNAs, sendo os mais relevantes descritos na sequência.

O Ensembl [26] é um banco de dados de vertebrados e outras espécies de eucariotos. Possui diversos tipos de ncRNAs anotados, dentre eles os lncRNAs. Sabe-se que as estruturas secundárias dos ncRNAs são muito variáveis, isto torna difícil detectar ncRNAs utilizando apenas sua sequência. Devido a isso, O Ensembl utiliza uma variedade de técnicas para detectar ncRNAs. Em primeiro lugar, uma combinação de pesquisas BLAST sensíveis são usados para identificar alvos prováveis, em seguida, uma pesquisa utilizando um modelo de covariância é utilizado para determinar a probabilidade de que os alvos podem dobrar-se em estruturas necessárias. Apresenta dados não muito acurados mas bons o suficiente quando se trata dos lncRNAs, sobre os quais não são conhecidas tantas informações. Por outro lado o Havana [28] aprensenta uma confiabilidade maior por

ser um banco de dados de modelos de genes de alta qualidade produzidos pela anotação manual dos genomas de vertebrados.

O DIANA Tools [18] tem o objetivo de fornecer algoritmos, banco de dados e software para interpretar e arquivar dados em uma estrutura sistemática. Ele possui dados de mRNAs e suas relações com lncRNAs. Podemos encontrar, também, bancos de dados especializados em lncRNAs, como é o caso do lncRNADisease [10], que disponibiliza informações, comprovadas experimentalmente, de lncRNAs que estão envolvidos em doenças, mostrando também o relacionamento desses com outros RNAs, DNAs e proteínas.

O lnCeDB [32] fornece uma base de dados de lncRNAs humanos que podem potencialmente atuar como ceRNAs (RNAs que compartilham elementos de reconhecimento de miRNA - MRE). Em lnCeDB além de procurar pares lncRNA-mRNA tem em comum miRNAs alvos, mas também comparar a expressão do par em 22 tecidos humanos para estimar as chances de o par de realmente estar ceRNAs.

Por fim, o LNCipedia [86] é um banco de dados para lncRNAS de humanos, transcritos e genes. Para informações básicas e sobre a estrutura do transcrito, várias estatísticas são calculados para cada entrada no banco de dados, tais como informações de estrutura secundária, a proteína que codifica locais potenciais e microRNA vinculativo. O banco de dados está disponível ao público e permite aos usuários consultar e baixar sequências e estruturas de lncRNA com base em diferentes critérios de pesquisa. A base de dados pode servir como uma fonte de informação sobre lncRNAs individuais ou como um ponto de partida para estudos de grande escala.

Capítulo 3

Aprendizagem de Máquina

Neste capítulo, conceitos básicos sobre aprendizegem de máquina serão apresentados, em particular, seus paradigmas e métodos computacionais de aprendizegem de máquina para extrair características de lncRNAs. Na Seção 3.1 os conceitos básicos de aprendizagem de máquina são descritos, bem como os seus paradigmas. Na Seção 3.2, a extração de características de lncRNA é definida. Para finalziar na Seção 3.3, os métodos computacionais SVM e Random-Forest são descritos.

3.1 Conceitos Básicos

Aprendizagem de Máquina é uma sub-área da Inteligência Artificial, que tem como principal foco a questão de como construir programas de computadores que automaticamente aprimoram-se com a experiência [55].

Um relatório recente do McKinsey Global Institute afirma que a aprendizagem de máquina (mineração de dados e análise preditiva) será o propulsor da próxima grande onda de inovação [51]. Nos últimos anos, muitas aplicações de grande sucesso utilizando aprendizagem de máquina foram desenvolvidas, tais como programas de mineração de dados, sitema de busca do Google, sistema de recomendação da Amazon, controle de tráfego por meio de radares de trânsito, reconhecimento facial, identificação e classificação de RNAs não codificadores (como é o caso deste trabalho) e muitos outros.

A aprendizagem de máquina ocorre quando programas aprendam a partir da experiência, adquirindo conhecimento de forma automática [55]. O aprendizado de máquina possui como principais vantagens sua a independência de domínio e a alta qualidade na predição. Os principais problemas relacionados a esses algoritmos são a necessidade de grandes quantidades de dados de treinamento e a necessidade de novos treinamentos com o advento de novos dados [1].

Existem quatro paradigmas de aprendizagem: não-supervisionada, supervisionada, por reforço e semi-supervisionada. Para cada uma destas técnicas foram desenvolvidos diversos algoritmos.

3.1.1 Aprendizagem Supervisionada

O processo de aprendizado supervisionado se dá pela apresentação de um conjunto de exemplos de treinamento rotulados a um indutor. A tarefa do indutor é então gerar

uma hipótese (classificador), também denominada descrição de conceito, tal que, dado um novo exemplo não rotulado, o classificador é capaz de predizer a sua classe [52].

A aprendizagem supervisionada tenta construir uma função que classifica objetos do conjunto de teste em uma das classes já conhecidas. A *performance* é calculada de acordo com o número de objetos do conjunto de teste classificados corretamente, levando em consideração os verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN).

Para problemas altamente desbalanceados, no entanto, a acurácia pode não fornecer informação adequada sobre a capacidade de discriminação de um classificador em relação a um dado grupo específico. Se um conjunto de dados apresenta uma classe minoritária correspondente a 2% das observações, um classificador com acurácia de 98% pode ser diretamente obtido por simplesmente classificar todo exemplo como pertencente à classe majoritária. Apesar de obter uma acurácia elevada, o classificador passa a ser inútil se o objetivo proposto for a identificação de exemplos minoritários [16].

O SVM *Support Vector Machine*, e o *Random Forest*, que serão discutidos mais a frente, são algoritmos que utilizam a aprendizagem supervisionada.

3.1.2 Aprendizagem Não-supervisionada

Aprendizagem não-supervisionada, por outro lado, permite abordar problemas com pouca ou nenhuma idéia de como resultados devem responder. Ocorre o reconhecimento de padrões em dados previamente não classificados para que cada dado de entrada seja agrupado em um conjunto específico de dados.

Nesta forma de aprendizagem, são descobertos relações, padrões, regularidades ou categorias nos dados que lhe são apresentados para serem codificados na saída. Um programa que somente utiliza técnicas de aprendizagem não-supersionada agrupa dados em classes, já que não tem informação de qual ação deve tomar e de qual estado é o desejado. Na aprendizagem não-supervisionada não há *feedback* com base nos resultados da previsão, pois não há nenhum mecanismo para corrigi-los.

O principal interesse do aprendizado não-supervisionado é desvendar a organização dos padrões existentes nos dados através de *clusters* (agrupamentos) consistentes. Com isso, é possível descobrir similaridades e diferenças entre os padrões existentes, assim como derivar conclusões úteis a respeito deles. Um *cluster* é uma coleção de objetos que são similares uns aos outros (de acordo com algum critério de similaridade pré definido) e dissimilares a objetos pertencentes a outros *clusters*.

O algoritmo de clusterização *K-means* é um algoritmo que utiliza a aprendizagem não-supervisionada.

Os principais algoritmos de clusterização são:

- Sequenciais:

  São algoritmos simples e rápidos, produzem como resultado um único agrupamento. Em sua grande maioria o resultado final depende da ordem em que tais dados são apresentados. Algoritmos caracterizados como sequenciais tendem a gerar agrupamentos compactos, na dependência da medida de distância usada. Esses algoritmos possuem a necessidade de um ou poucos passos onde o número de grupos não é conhecido inicialmente e, geralmente, têm como entrada um valor que determina o

número máximo de grupos a serem criados. Leva-se em consideração o valor máximo de grupos, associado a essa distância, para ser feito um cálculo de distância apropriado os dados aos grupos, para definir os grupos de cada dado [68].

- Hierárquicos:

  Duas abordagens podem ser derivadas do clustering hierárquico: aglomerativo (*Botton-up*) e divisivo (*Top-down*). Na primeira abordagem, os dados são inicialmente distribuídos de modo que cada exemplo represente um *cluster* e, então, esses *clusters* são recursivamente agrupados considerando alguma medida de similaridade, até que todos os exemplos pertençam a apenas um cluster. Na segunda abordagem, o processo inicia-se com apenas um agrupamento contendo todos os dados e segue dividindoo recursivamente segundo alguma métrica até que alcance algum critério de parada, frequentemente o número de *clusters* desejados [5].

  A Figura 3.1 apresenta um exemplo de árvore de *clusters* na clusterização hierárquica.

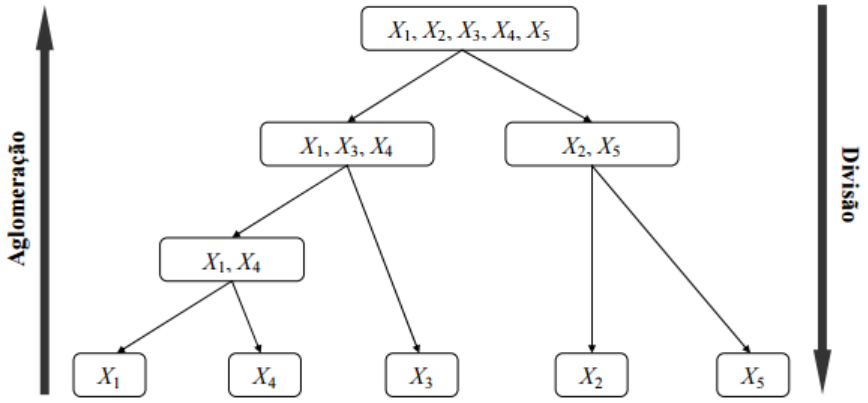


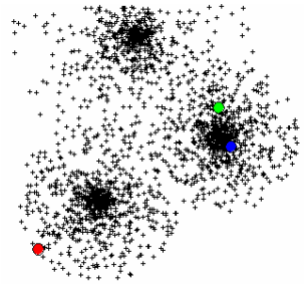
Figura 3.1: Exemplo de Árvore de *clusters* na clusterização hierárquica [60].

K-means

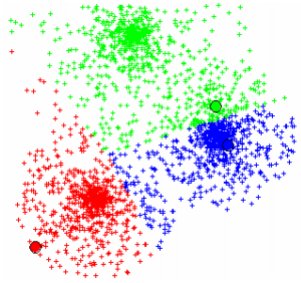
O *K-means* é um dos mais simples algoritmos de aprendizagem não supervisionada que resolvem o problema de agrupamento. *K-means* é uma técnica que usa o algoritmo de agrupamento de dados por K-médias. O objetivo deste algoritmo é encontrar a melhor divisão de N dados em K grupos de maneira que a distância total entre os dados de um grupo e o seu respectivo centro, somada por todos os grupos, seja minimizada. A idéia principal é definir centróides k, um para cada *cluster*. Estes centróides devem ser colocados de uma forma astuta por causa da localização diferente gera um resultado diferente. Portanto, a melhor escolha é colocá-los tanto quanto possível longe um do outro.

O centro do *cluster* inicial é formado para cada caso em torno dos dados mais próximos e, então, são comparados com os pontos mais distantes e os outros *clusters* formados. Por meio de um processo de atualização contínua e de um processo interativo um ciclo é

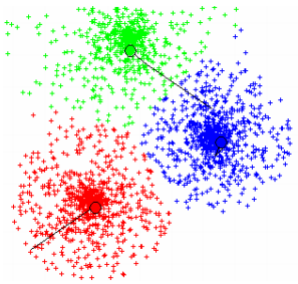
gerada. Como resultado deste ciclo podemos notar que as k centróides mudam o seu nível de localização a passo até que não haja mais mudanças a serem feitas. Em outras palavras centroids não se movem mais. Dessa forma os centros dos *clusters* finais são encontrados. O funcionamento do *K-means* pode ser melhor compreendido na Figura 3.2.



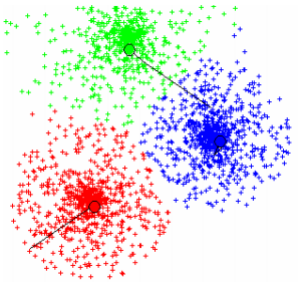
(a) 3 centróides gerados randomicamente.



(b) Atribui a cada um dos objetos o grupo que tem o centróide mais próximo.



(c) Posições dos centróide recalculados.



(d) Todos os pontos foram agrupados.

Figura 3.2: Etapas do *K-means* [82].

3.1.3 Aprendizagem Semi-supervisionada

Aprendizagem semi-supervisionada tornou-se, recentemente, uma boa alternativa para aumentar a capacidade de generalização de modelos de aprendizagem de máquina [17]. No domínio da aprendizagem de máquina, a aprendizagem semi-supervisionada ocupa o meio termo, entre a aprendizagem supervisionada (na qual todos os exemplos de treinamento são rotulados) e a aprendizagem não supervisionada (em que os dados não são rotulados).

Este paradigma é útil em casos onde o conjunto de treinamento não fornece informação suficiente para a indução de uma regra-geral. Assim, utiliza-se o conjunto de teste como fonte extra de informação para a resolução do problema.

Dentre os problemas em que essa abordagem é útil estão todos aqueles onde o espaço de amostragem é muito grande para ser possível gerar uma amostra estatisticamente representativa, ou ainda nos casos em que o ocorra um alto grau de especialização do classificador ou que o mesmo possua um o custo computacional caro [72].

O interesse na aprendizagem semi-supervisionada aumentou nos últimos anos, particularmente devido a domínios de aplicação em que os dados não rotulados são abundantes, como imagens, texto e bioinformática.

Através da abordagem semi-supervisionada é possível minimizar os dados ruidosos do conjunto de treinamento, melhorando os resultados obtidos [17].

As Máquina de Vetores de Suporte Transdutoras (*Transductive Support Vector Machine - TSVM*) é um exemplo de algorítmo que utiliza a aprendizagem semi-supervisionada.

A TSVM é a inferência transdutiva da Máquina de Vetores de Suporte (SVM). O objetivo da aprendizagem transdutiva é inferir os rótulos corretos apenas para os dados não rotulados inicialmente. O TSVM utiliza as informações transportadas pelas amostras não rotulados para classificação e adquire um melhor desempenho de classificação do que a SVM regular.

3.1.4 Aprendizagem por Reforço

Aprendizado por Reforço pode ser visto como uma forma de programar agentes utilizando recompensas e punições para resolver tarefas específicas através de interações com o ambiente [44]. O Aprendizado por Reforço não é definido como um conjunto de algoritmos de aprendizagem. O Aprendizado por Reforço é uma classe de problemas de aprendizagem. Todo o algoritmo que resolve bem esse problema é considerado um algoritmo de aprendizado por reforço [76].

O programa percebe e interage com o ambiente, o qual é caracterizado por todos os outros elementos, exceto o programa (Agente). As ações tomadas pelo programa geram recompensas (Reforço), sendo que essas recompensas dizem qual a melhor ação a ser tomada, dados os possíveis estados do ambiente conhecidas [76]. O papel do aprendizagem por reforço é usar recompensas obtidas para aprender qual ação é ótima, ou próxima da ação ótima, em determinado ambiente [55]. A Figura 3.3 como o fluxo de ações no ambiente geram as recompensas e como as recompensas no agente geram as futuras ações.

Figura 3.3: Diagrama do funcionamento da aprendizagem por Reforço [82].

3.2 Extração de características

Para anotar e classificar ncRNAs precisa-se de características que possam ser utilizadas nos métodos computacionais. São essas características que permitem por indicar a qual família de ncRNAs uma determinada molécula pertence. Mas quais características são importantes para se classificar uma molécula de RNA? A resposta a essa pergunta é simples, quando se observa um exemplo mais próximo do nosso dia a dia. Para um ser humano por exemplo, o que o define um ser humano e não um outro animal? Sua resposta

poderia ser o fato de nós, seres humanos, sermos animais bípedes. Sim, sua observação foi correta e relevante visto que todos os animais quadrúpedes estariam fora de questão nesta análise. Observa-se, porém, aspectos únicos e exclusivo dos seres humanos como a habilidade da fala, escrita dentre outras. Essas habilidades humanas seriam, então, ótimos indícios para se classificar um animal como sendo humano ou não.

Esta mesma filosofia pode ser posta em prática quando estudamos ncRNAs. O caso dos ncRNAs longos, por exemplo, possuem como características importantes para sua classificação os fatos de não apresentarem ORFs suficientes para codificar proteínas e de possuírem um comprimento de mais de 200 nucleótidos.

RNAs não-codificadores longos intergênicos (lincRNAs), que foram apresentados na Seção 2.2.1, foram catalogados para humanos, camundongos, peixe-zebra, sapos e outras espécies [83]. Essa catalogação se dá pelo uso de modelos que consideram diversas características como: posição de início no genoma, posições de *splicing* e posição da cauda poli-A de cada transcrito. Na Figura 3.4, podemos ver diversos métodos utilizados para identificar lincRNAs em humanos e camundongos.

Busca-se então, por meio de processos computacionais, a identificação de tais características relevantes a classificação de uma determinada molécula de ncRNA. Alguns métodos que utilizam aprendizagem de máquina para classificação e extração de características são mostrados na Seção 3.3.

3.3 Métodos

Nesta Seção são descritos em detalhes o método SVM [40], que será utilizado para criação de um modelo preditivo e o método *Random Forest* [9], que além de criar um modelo preditivo também será utilizado para a extração de características dos lncRNAs.

3.3.1 SVM

Uma máquina de vetores de suporte (*Support Vector Machine - SVM*) é uma máquina linear que tem como principal tarefa, no contexto de problemas de classificação de padrões, construir um hiperplano como superfície de decisão, de tal modo que a margem de separação entre amostras positivas e negativas é maximizada [35]. As SVMs podem ser utilizadas tanto para classificação quanto para regressão [35] adquirindo com o aprendizado na etapa de treinamento a capacidade de generalização.

SVMs são baseados no princípio estrutural de Minimização de Risco [84] da teoria da aprendizagem computacional. A ideia de minimização do risco estrutural é encontrar uma hipótese h para o qual podemos garantir o menor erro verdadeiro. O verdadeiro erro de h é a probabilidade de que h fará um erro em um invisível e aleatoriamente selecionado caso de teste. Um limite superior pode ser utilizado para ligar o verdadeiro erro de uma hipótese h com o erro de h no conjunto de treinamento e a complexidade de H (o espaço contendo a hipótese h) [84].

Para um caso binário como é mostrado na Figura 3.5, o objetivo da SVM é separar as instâncias das duas classes através de uma função que será obtida a partir dos exemplos conhecidos na fase de treinamento. O objetivo é produzir um classificador que funcione de forma adequada com exemplos não conhecidos, ou seja, exemplos que não foram aplicados

Table 1. Large-Scale Efforts to Catalog lincRNA Loci and Transcripts

Reference	Data for Transcript Reconstruction	Genomic Features and Filters	Coding-Potential Filters	Number of lincRNAs
Mouse				
Ravasi et al., 2006	cDNAs		Manual curation, ORF length, CRITICA	13,502 transcripts
Ponjavic et al., 2007	cDNAs, CAGE		Manual curation, ORF length, BLAST, CRITICA	3,122 transcripts
Guttman et al., 2009	Chromatin marks, tiling arrays	Collection of approximate exonic regions, chromatin domain ≥5 kb	CSF	1,675 loci (1,250 conservatively defined)
Guttman et al., 2010	RNA-seq	Multi-exon only	CSF	1,140 lincRNA transcripts
Sigova et al., 2013	RNA-seq, cDNAs, chromatin marks,	Antisense overlap with mRNA introns allowed, ≥100 nt mature length	CPC	1,664 loci
Human				
Khalil et al., 2009	Chromatin marks, tiling arrays	Collection of approximate exonic regions, chromatin domain ≥ 5 kb	CSF	3,289 loci
Jia et al., 2010	cDNAs	Overlap with mRNAs allowed		5,446 transcripts
Ørom et al., 2010	cDNAs	Restricted to loci >1 kb away from known protein-coding genes, ≥200 nt mature length	Manual curation based on length, conservation and other characteristics of the ORFs	3,019 transcripts from 2,286 loci
Cabili et al., 2011	RNA-seq	Multi-exon only, ≥200 nt mature length	PhyloCSF, Pfam	8,195 transcripts (4,662 in the stringent set)
Derrien et al., 2012	cDNAs	Overlap with mRNAs allowed (intergenic transcripts reported separately), ≥200 nt mature length	Manual curation based on length, conservation and other characteristics of the ORFs	14,880 transcripts from 9,277 loci, including 9,518 intergenic transcripts
Sigova et al., 2013	RNA-seq, cDNAs, chromatin marks,	Antisense overlap with mRNA introns allowed, ≥100 nt mature length	CPC	3,548 loci from embryonic stem cells, and 3,986 loci from endodermal cells

Figura 3.4: Métodos utilizados para classificação de lincRNAs em humanos e camundongo [83].

durante o treinamento, adquirindo assim a capacidade de predizer as saídas de futuras novas entradas. Uma SVM constrói um classificador de acordo com um conjunto de padrões por ele identificados nos exemplos de treinamento, onde a classificação é conhecida.

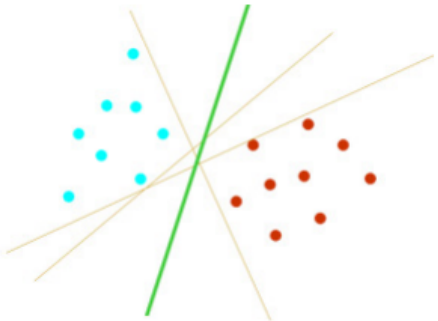


Figura 3.5: Hiperplano de máxima margem de separação [42].

É possível que, para a o exemplo da Figura 3.5, existam vários classificadores lineares que separam essas duas classes, mas apenas um será o que maximiza a margem de separação (distância da instância mais próxima ao hiperplano que separa as duas classes). O hiperplano com margem máxima é chamado de hiperplano ótimo, que será o objeto de busca do treinamento do classificador. A Figura 3.6 faz uma comparação entre o hiperplano ótimo e hiperplano de margem pequena.

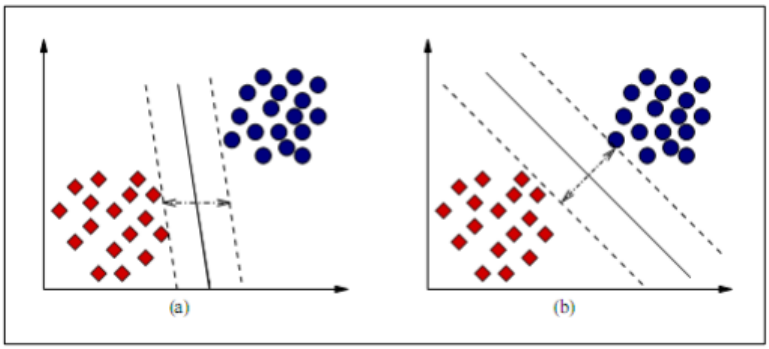


Figura 3.6: (a) Hiperplano com margem pequena de separação (b) Hiperplano com margem máxima de separação [42].

3.3.2 Métodos de Aprendizagem *Ensemble*

O objetivo destes tipos de métodos é o de combinar as previsões de vários estimadores de base construídas com um determinado algoritmo de aprendizagem, a fim de melhorar a generalização e robustez em vez de utilizar um único estimador.

Bagging

Neste método que é uma abreviação para *Bootstrap Aggregation* é a maneira diminuir a variância de uma predição através da geração de dados adicionais para treinamento a partir dos dados originais usando combinações com repetições para produzir subconjuntos de mesmo tamanho (número de instâncias) de seus dados originais. Os valores das predições individuais de cada subconjunto são agregadas para uma predição final.

O resultado da geração do método *Bagging* é um conjunto de classificadores que são utilizados de forma integrada, pois cada nova instância a ser classificada será avaliada pelo classificador composto cujo resultado (a classificação da instância) será a resposta escolhida pela maioria dos k classificadores. As principais características no processo de geração de classificadores são:

- Geração de diferentes amostras de tamanhos iguais a partir da mesma base de dados de treinamento;
- Obtenção de um classificador para cada amostra;
- Ocorre assim a obtenção de um classificador composto que inclui todos os classificadores individuais gerados na fase anterior e por meio de uma votação simples é escolhida a classificação mais popular dentre os classificadores individuais.

Ao aumentar o tamanho do seu conjunto de treinamento, não se pode melhorar a força preditiva do modelo, mas apenas diminuir a variância. Como exemplo, o algoritmo *Random Forest* combina árvores de decisão aleatórios utilizando o *Bagging* para conseguir diminuir sua variância para classificação [8].

Boosting

O *Boosting* tem como objetivo melhorar a precisão de qualquer algoritmo de aprendizagem. Como no método *Bagging*, são geradas amostras que dão origem a classificadores que são utilizados de forma integrada. Diferem na maneira como são geradas as amostras e de como são combinados os resultados dos classificadores. Em vez de gerar amostras aleatoriamente a partir de um dado original, levam-se em conta as amostras já geradas de forma a alterar a distribuição de geração das próximas amostras.

A geração de k amostras no método *Boosting* consiste em:

- Gerar a primeira amostra com uma distribuição uniforme, ou seja, todas as instâncias da base têm a mesma probabilidade (1/n) de serem incluídas na primeira amostra gerada;[47];
- Gerar um classificador para esta amostra e aplicar o classificador a base de treinamento original [47];
- Diminuir de acordo com o classificador gerado, a probabilidade de serem incluídas na próxima amostra das instâncias que foram corretamente classificadas e aumentar a probabilidade das instâncias que foram incorretamente classificadas [47];
- Gerar a segunda amostra a partir da base original levando em conta as novas probabilidades de cada instância [47];
- Gerar um classificador para a segunda amostra e aplicar o classificador a base de treinamento original [47];
- Diminuir a probabilidade das instâncias bem classificadas e aumentar a probabilidade das instâncias mal classificadas [47];
- Repetir este processo de geração de amostras, classificadores e alteração de pesos até serem gerados a k-ésima amostra e o k- ésimo classificador [47].

Como o peso é maior para exemplos classificados incorretamente, a probabilidade desse elemento ser escolhido para o próximo subconjunto de treinamento é grande.

Ao invés de uma votação simples entre as respostas fornecidas por cada classificador, no método *Boosting* a votação é ponderada segundo um índice de importância entre os classificadores gerados. Isto pode ser realizado pois ocorre a memorização da eficiência de cada classificador gerado frente à base de treinamento. Ao combinar as vantagens e desvantagens dessas abordagens, variando a sua fórmula de ponderação pode-se ter uma boa força preditiva para uma ampla gama de dados de entrada.

3.3.3 *Random Forest*

Esta Seção apresenta o *Random Forest* que será o algoritmo utilizado neste projeto. É um método que utiliza técnicas que são potencialmente capazes de identificar variantes

onde o modelo causal é desconhecido e de lidar com o problema da dimensionalidade dos dados. O *Random Forest* foi o algoritmo escolhido por apresentar características importantes como sua simplicidade, flexibilidade, escalabilidade e capacidade de lidar com um grande número de variáveis de entrada sem incorrer de sobre-ajuste [2].

O *Random Forest* integra um conjunto de métodos de aprendizado de máquina que envolve da construção de muitos preditores (classificadores ou regressores) e cuja predição consiste na agregação das predições de todos os preditores do conjunto. Para a criação desse método que foi proposto por Breiman [8] o mesmo utilizou-se de seus trabalos passados sobre as árvores CART (*Classification and Regression Tree*) e *bootstrap and aggregating* (*Bagging*) além dos trabalhos que utilizaram árvores aleatórias para a solução de problemas de classificação.

O *Random Forest* é uma combinação de árvores de decisão as quais são geradas para serem utilizadas na classificação de novas Classes. O *Random Forest* apresenta excelentes características de precisão, generalização para outras amostras que não aquelas em que o classificador foi treinado e capacidade de bom desempenho em pequenas amostras. O erro de generalização para as florestas converge a um limite quando o número de árvores na floresta se torna grande. O erro de generalização de uma floresta depende da caracteristica das árvores individuais na floresta e a correlação entre elas [8].

Árvores de Decisão

As árvores de decisão são representações simples de forma gráfica de decisões e suas possíveis consequências. São um meio eficiente de minerar classes e várias outras informações extremamente úteis que são extraídas em valores de atributos de conjuntos de dados. Em outras palavras, uma árvore de decisão nada mais é do que um mecanismo que auxilia na tomada de decisões.

Pode ser utilizada para alcançar um objetivo que por meio de regras de decisções dividindo sucessivamente uma grande coleção de dados em conjuntos menores (subconjuntos). Uma árvore de decisão é um modelo preditivo uma vez que faz um mapeamento de observações sobre um item para conclusões sobre o seu valor esperado. Os nós internos da árvore correspondem a uma variável. O valor que acompanha a aresta de ligação a seu filho corresponde a um possível valor dessa variável; Uma folha correseponde ao valor previsto para a variável após tomar todas as decisões ao longo do caminho desde a raiz. A Figura 3.10 explica bem o funcionamento de uma árvore de decisão.

Classificação utilizando árvores de decisão

É utilizado um algoritmo de aprendizado de máquina para construir um modelo de classificação. O processo consiste na seleção de um classificador que será utilizado na predição as classses desconhecidas. Na montagem desse modelo os valores das classes dos exemplos do conjunto de treinamento são conhecidas. Com o modelo construido esse classificador pode ser utilizado para predizer as classses do conjunto teste, onde as classes são desconhecidas.

O principal motivo para se utilizar árvores de decisão em problemas de classificação é o fato do conhecimento adquirido ser representado por meio de regras. Na construção da árvore associa-se a cada nó o atributo de maior relevância, entre todos os outros atri-

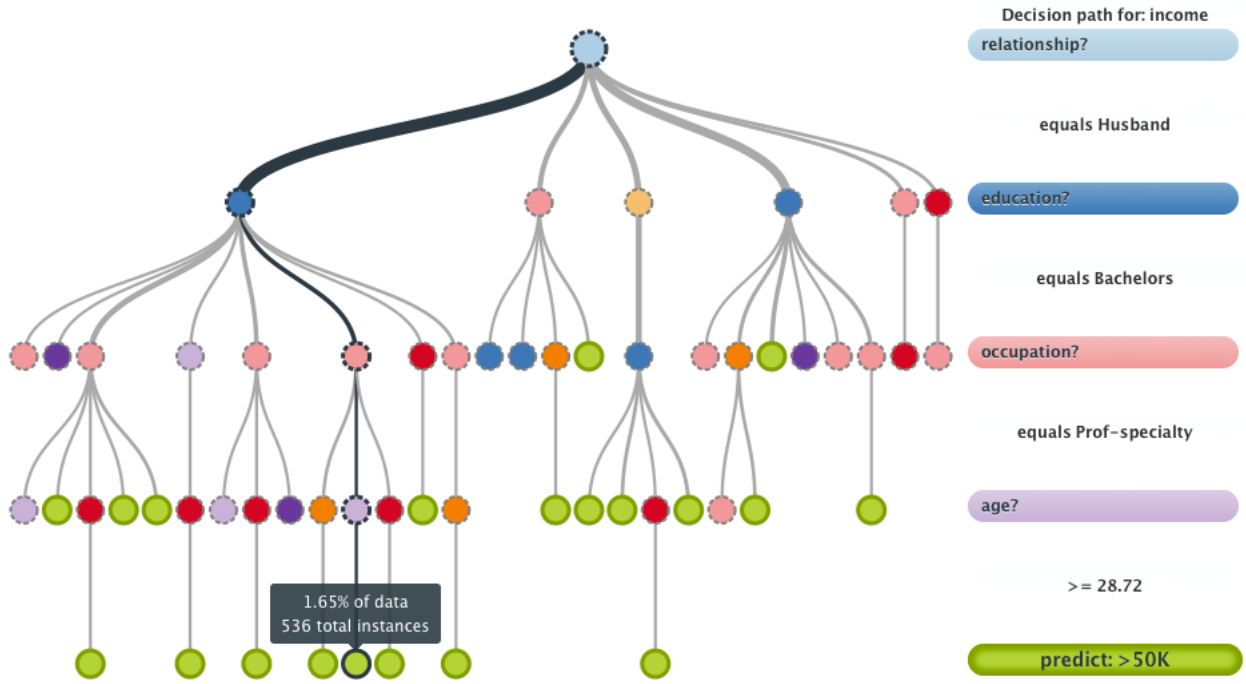


Figura 3.7: Funcionamento de uma Árvore de Decisão [19].

butos não utilizados até então. Algoritmos que implementam árvores de decisão possuem diferentes técnicas para determinar a importância dos atributos em relação aos outros.

A busca por atributos que melhor dividem o conjunto de dados exemplos em sub-conjuntos é feita de forma recursiva e por meio de uma busca gulosa. No início os exemplos são colocados na raíz da árvore. De forma recursiva um atributo preditivo é escolhido para representar o teste desse nó e, assim, dividir os exemplos em sub-conjuntos de exemplos. O processo é repetido até que todos os exemplos estejam classificados ou então até que todos os atributos preditivos tenham sido utilizados.

Escolha dos atributos preditivos para os nós da Árvore

Os critérios de seleção dos atributos preditivos a serem utilizados em cada nó, são definidos em termos da distribuição de classe dos exemplos antes e depois da divisão [78]. O algoritmo tentará encontrar o melhor atributo para realizar essa divisão de forma que cada nó interno da árvore é dividido de acordo com um único atributo.

Diferentes medidas, tais como impureza, distância e dependência são utilizadas na seleção dos atributos que causam a melhor divisão. Divide-se os dados de um nó-pai de forma a minimizar o grau de impureza dos nós-filhos. Quanto menor o grau de impureza, mais desbalanceada é a distribuição de classes. A impureza é nula quando todos os exemplos de um nó pertencerem à mesma classe, enquanto a impureza é máxima quando há o mesmo número de exemplos para cada classe possível de um nó [43].

O Ganho de Informação é uma das medidas baseadas em impureza, o qual usa a entropia como medida de impureza [43]. O ganho de informações escolhe uma divisão com base no atributo mais informativo.

Para determinar o quão boa seria uma divisão em um determinado atributo, é necessário comparar o grau de entropia do nó gerador com o grau de entropia dos nós gerados. O atributo que gerar uma maior diferença é escolhido para o determinado nó [43]. Sendo a entropia a medida da impureza das amostras dos exemplos de treinamento S pode-se calcular a entropia por meio da Equação 3.1:

$$Entropia(S) = -p \log_2 p - n \log_2 n \tag{3.1}$$

Onde p é a porção de exemplos positivos em S enquanto n a de exemplos negativos. Essa Equação pode ser expandida para a Equação 3.2:

$$Entropia(S) = \sum_{i} -Pi \log_2 Pi \tag{3.2}$$

Onde a entropia de uma variável nominal S pode tomar i valores.

O ganho de informação passa a ser então apresentado por meio da Equação 3.3:

$$Ganho = Entropia(pai) - \sum_{j=1}^{n} \left[\frac{N(Pj)}{N} Entropia(Pj) \right] \tag{3.3}$$

Onde n é o número de valores do atributo, ou seja, o número de nós-filhos, N é o número total de objetos do nó-pai e *N(Pj)* é o número de exemplos associados ao nó-filho Pj [43].

Logo, é selecionado pelo algoritmo o atributo que maximiza o ganho de informação. Embora esse método ofereça bons resultados, ele favorece a divisão em variáveis que possuem um grande número de atributos. Este problema ocorre por exemplo, ao se utilizar um atributo totalmente irrelevante, como um identificador único de forma que um nó seria criado para cada valor possível, onde o número de nós seria igual ao número de identificadores. Dessa forma cada um desses nós possuiria apenas um elemento, pertecente a uma única classe, ou seja, os exemplos seriam totalmente discriminados. De forma que, o valor da entropia seria mínima tendo em vista que, em cada nó, todos os elementos (neste caso único) pertencem à mesma classe. Essa divisão geraria um ganho máximo, embora seja totalmente inútil [43].

Para solucionar o problema do ganho de informação, foi proposto em Quinlan [67] a Razão de Ganho (*Gain Ratio*), que nada mais é do que o ganho de informação relativo (ponderado) como critério de avaliação. A razão do ganho pode ser definida como apresentado na Equação 3.4:

$$RazaoDeGanho(no) = \frac{Ganho}{Entropia(no)} \tag{3.4}$$

Razão de ganho de informação é uma razão de ganho de informação para a informação intrínseca. Ele é usado para reduzir um viés em relação a atributos com vários valores, levando em conta o número e o tamanho dos ramos ao escolher um atributo A razão não é definida quando o entropia é igual a zero. Além disso, a razão de ganho favorece atributos cujo a entropia, possui valor pequeno.

Para solucionar esse problema Quinlan [66] sugere que primeiro seja calculado o ganho de informação para todos os atributos. Após isso, são selecionados os atributos que obtiveram um ganho de informação acima da média. Por fim, escolhe-se o atributo que possui a melhor razão de ganho.

Existe também a medida Gini, a qual gera um índice de dispersão estatística. Este índice mede a heterogeneidade dos dados. O índice de Gini é calculado subtraindo a soma das probabilidades quadradas de cada classe por um. Favorece partições maiores Para um problema de c classes, o *gini index* é definido segundo a Equação 3.5:

$$gini\_index(n) = 1 - \sum_{i=1}^{c} p(i/n) \tag{3.5}$$

O nó é puro quando este índice é igual a zero. De forma análoga, o nó é impuro quanto mais próximo de 1, pois ocorre o aumento do número de classes uniformemente distribuídas no nó.

Como no cálculo do ganho de informação, é calculado a diferença entre o *gini index* antes e após a divisão. Dessa forma, Gini, é calculado segundo a Equação 3.6:

$$Gini = gini\_index(pai) - \sum_{j=1}^{n} \left[\frac{N(Pj)}{N} gini\_index(Pj) \right] \tag{3.6}$$

Onde n é o número de valores do atributo, ou seja, o número de nós-filhos, N é o número total de objetos do nó-pai e *N(Pj)* é o número de exemplos associados ao nó-filho *Pj* [43].

Quando, o critério de Gini é utilizado tende-se a isolar num ramo os dados que representam a classe mais frequente.

Funcionamento do *Random Forest*

No *Random Forest* para a construção de cada árvore do modelo é utilizado o método ensemble *Bagging* que consiste na criação de uma amostra retirada com substituição de um conjunto de treinamento. A decisão escolhida em um nó durante a montagem da árvore passa a ser, então, a melhor escolha de um subconjunto aleatório da amostra. A classificação elegida é aquela que for a mais votada dentre todos os subconjuntos. Utilizando-se a aleatoriedade para uma floresta, comparado a uma única árvore, há um aumento na polarização enquanto ocorre uma redução da variância devido ao calculo da média [2]. Como a redução da variância é maior do que o aumento da polarização a aleatoriedade nos proporciona um melhor modelo preditivo.

O *Random Forest* utiliza a árvore CART como preditor base para sua construção. A raiz da árvore CART representa o conjunto de todos os atributos, cada nó da árvore possui uma variável preditora a qual particiona o conjunto de atributos em outros dois subconjuntos. O processo é repetido sucessivamente até que ao final do processo obtenha-se uma árvore binária, a qual é utilizada para fazer predições por meio de um processo de busca. Um caso teste irá percorrer em cada árvore da floresta seus possíveis caminhos de acordo com as decisões tomadas em cada nó, a variável preditora de cada nó determina se a busca prossegue pelo ramo direito ou esquerdo até que se encontre um nó folha, o qual determina a predição do caso teste. Para casos de classificação a predição mais votada de

todas as árvores é escolhida como predição final, já para casos de regressão a predição é uma média dos valores do caso teste [2].

O *Bagging* é uma técnica para construção de conjuntos de preditores, construídos sucessivamente de forma independente, utilizando uma amostra *bootstrap* do conjunto de dados de treinamento [2]. Comparado com o preditor base métodos que utilizam *Bagging* apresentam um menor erro de predição por meio da redução do componente de variância do erro. Porém, essa redução é limitada pela correlação entre os preditores. Para solucionar esse problema o *Random Forest* porpõe mais uma forma de aleatoriedade para obter preditores menos correlacionados. Ao gerar cada nó de uma árvore apenas parte das variáveis disponíveis são selecionadas para determinar a melhor partição [2].

A Figura 3.8 mostra o algoritmo *Random Forest* proposto por Breiman onde *ntree* é o número de árvores na floresta e o *mtry*, o número de variáveis utilizadas para particionar os nós das árvores [2].

Algoritmo 1: Random Forest (ntree, mtry)

```
1  para b ← 1 ... ntree faça
2      selecionar uma amostra bootstrap;
3      repita
4          selecionar aleatoriamente mtry variáveis;
5          encontrar o resultado das melhores partições;
6      até formar a árvore;
7      predizer Y para OOB;
8      predizer Y para X permutado;
9  fim para
10 calcular o erro OOB;
11 calcular a importância das variáveis;
```

Figura 3.8: Algoritmo *Random Forest* [2].

O *Random Forest*, é capaz de estimar tanto o erro de predição quanto a importância das variáveis analisadas. Essas estimativas são geradas avaliando os dados *out-of-bag* (OOB) que são as amostras do conjunto de dados de treinamento que não foram incluídas no conjunto das amostras bootstrap, que em média representam 36% das amostras de treinamento [2].

O *Random Forest* utiliza cada árvore construída para predizer os valores dos dados OOB. Essas predições são comparadas com os verdadeiros valores para obter uma estimativa de erro, chamado erro OOB. Os dados OOB também são utilizados para identificar a importância das variáveis. Compara-se com o erro OOB aos valores dos erros quando se permuta cada uma das variáveis utilizadas na construção de cada árvore. A importância de uma variável é medida pelo impacto que a retirada de sua informação causa no erro de predição OOB [2]. A Figura 3.9 representa os procedimentos para a estimativa do erro OOB e a identificação da importância das variáveis.

Em resumo para construir uma floresta é necessário:

1. Aleatoriamente se cria um subconjunto com substituição (*Bagging*) de tamanho N a partir de um conjunto original sendo esse subconjunto uma árvore da floresta. Alguns dados podem ser escolhidos mais de uma vez e outros nunca serem escolhidos onde as chances de um dado estar em um subconjunto é de 66% ;

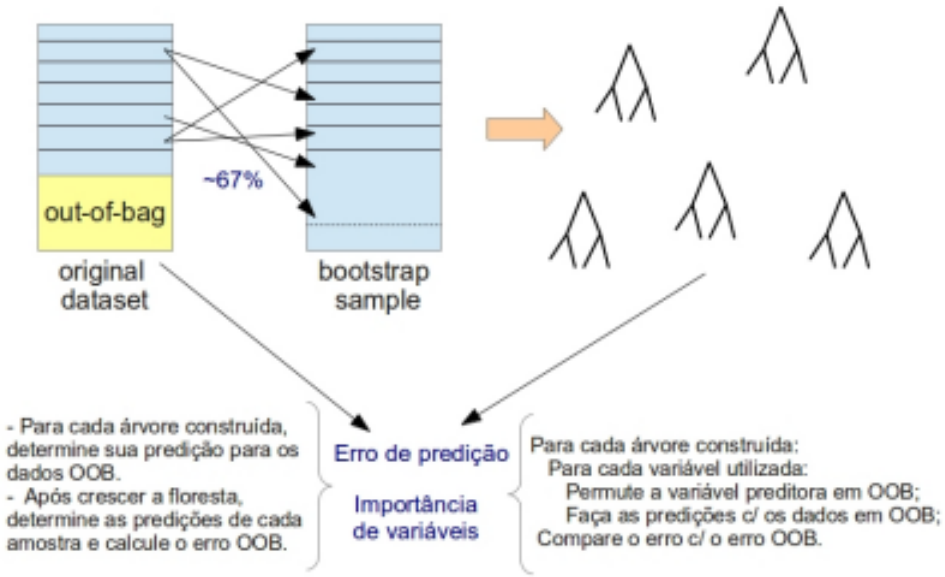


Figura 3.9: Procedimento embutido em RF para estimar o erro OOB e da importância das variáveis [2].

2. Para cada subconjunto selecionado é gerado uma árvore de decisão repetindo recursivamente os passos listados abaixo para cada nó da árvore até que o nó de tamanho mínimo seja encontrado.
   - São selecionados n características das N existentes;
   - É selecionada a melhor decisão;
   - Divide o nó em dois nós filhos;
3. Gera a saída que é o conjunto de árvores que serão utilizadas para geração de uma média.

A Figura 3.10 demonstra um exemplo do funcionamento do *Random Forest* para a criação de uma árvore em um caso binário.

Assim, observa-se na Figura 3.10 que para a separação das classes são gerados hiperplanos que são determinados por meio do atributo escolhido dentre todas aquelas utilizadas como parâmetros de entrada no classificador. Sendo assim, o classificador *Random Forest* separa as superfícies de decisão por meio da criação de uma sequência de hiperplanos paralelos aos eixos [13].

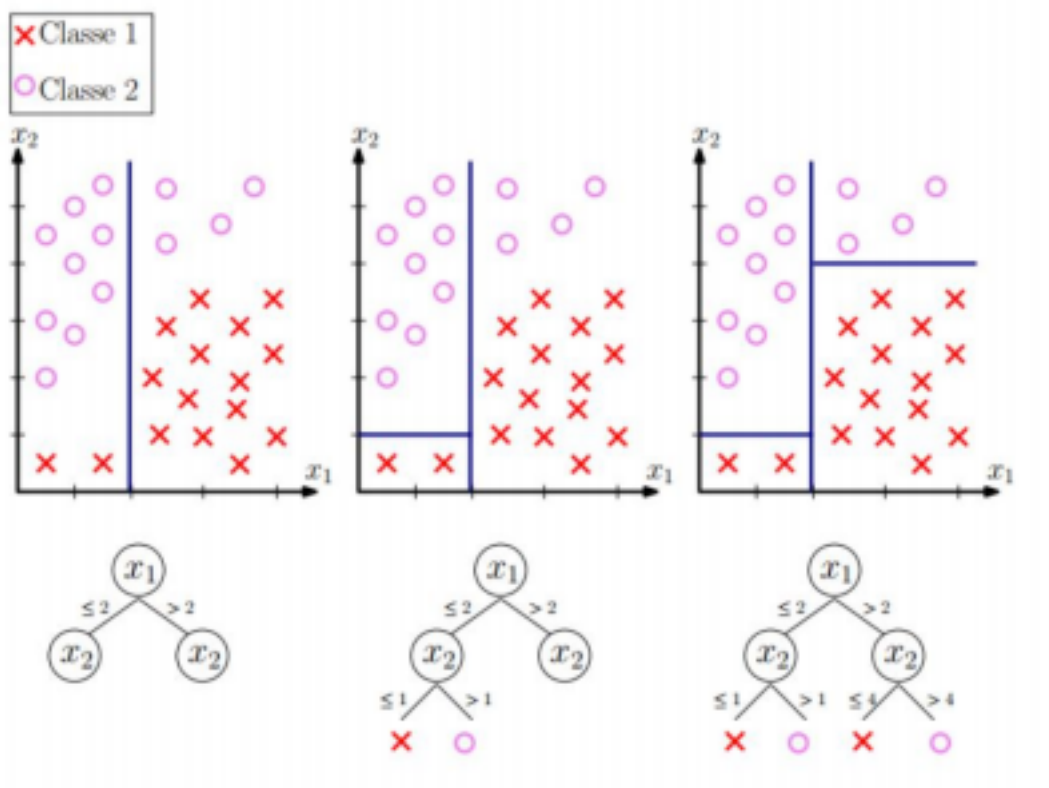


Figura 3.10: Processo de construção de uma árvore de decisão no *Random Forest* [48].

Capítulo 4

Projeto de Extração de Características

Neste capítulo é descrito o projeto de extração de características para lncRNAs. Na Seção 4.1, o método de extração de características é descrito. Na Seção 4.2, os testes a serem realizados para análise das características são apresentados. Por último, na Seção 4.3, detalhes de como o projeto foi implementado são mostrados.

4.1 Descrição do método

Nesta Seção, inicialmente descreve-se o método proposto para extrair características que podem ser utilizadas para predizer lncRNAs (veja Figura 4.1). As etapas descritas nessa Figura 4.1 são descritas a seguir.

Inicialmente devemos obter uma base de dados de lncRNAs para serem utilizados nos modelos preditivos. Em seguida, precisamos definir características de lncRNAs que podem ser importantes para sua classificação, atualmente sabe-se que o tamanho da ORF é importante para a classificação dos lncRNAs, pois esses apresentam baixo potencial de codificação, logo possuem ORFs pequenas.

Depois de selecionar as características que serão utilizadas, deve-se criar um conjunto de dados de treinamento e teste com dados positivos e negativos, respectivamente, lncRNAs e PCTs (*Protein Coding Transcripts*). Em seguida, deve-se utilizar o algoritmo *Random Forest* e SVM para gerar um modelo preditivo com o conjunto de dados de treinamento obtido na etapa anterior. Após a criação do modelo preditivo, deve-se utilizar o *Random Forest* para determinar as principais características do modelo que determinam se uma sequência é lncRNA ou não. Em seguida, deve-se avaliar a *performance* do modelo preditivo utilizando o conjunto de dados de teste em dois métodos de aprendizagem de máquina, *Random Forest* e SVM, respectivamente.

É preciso comparar os resultados obtidos com resultados encontrados na literatura que utilizaram algoritmos e técnicas diferentes. Deve-se, por último, determinar o grau de confiabilidade dos resultados. Para isso, deve-se comparar o desempenho dos modelos preditivos nos métodos de aprendizagem de máquina *Random Forest* e SVM quando as características apontadas como as mais importantes são utilizadas.

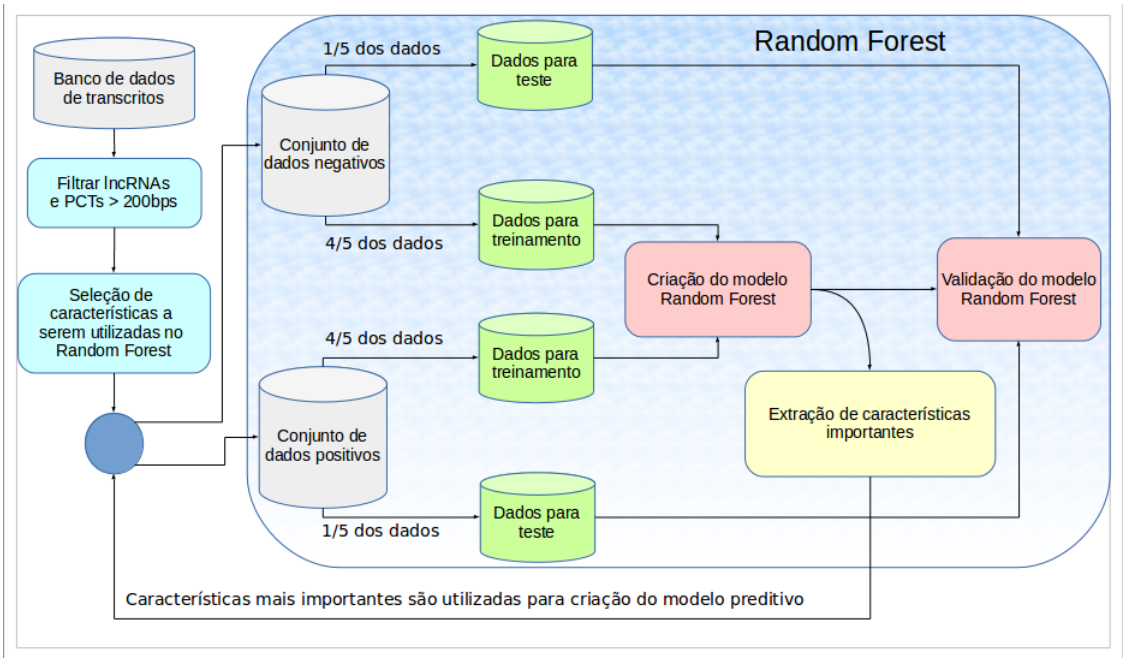


Figura 4.1: Fluxo do projeto de extração de características utilizando o *Random Forest*.

4.1.1 Características

O projeto consiste na extração de características relevantes para determinar se uma determinado transcrito é um lncRNA. Para isso, este projeto usou apenas a estrutura primária das moléculas de ncRNAs e transcritos codificadores de proteínas, ou seja, sua sequência de bases nitrogenadas.

Para esse projeto foram selecionadas 345 características. Essas características foram divididas em três grandes grupos (veja Figura 4.2). O primeiro grupo leva em conta o potencial de codificação de cada transcrito. Esse potencial foi calculado pela proporção dada pelo tamanho da ORF dividido pelo tamanho da sequência. As primeiras, menores e maiores ORFs de cada transcrito foram selecionadas para fazer parte deste grupo. O segundo grupo consite nas posições de início e fim das ORFs. Para isso, foram selecionados de cada transcrito as posições de início e fim da primeira, menor e maior ORF. O terceiro grupo de dados agrupa as 336 características restantes, que são as frequências relativas médias de todos os di, tri e tetra-nucleotídeos de um transcrito.

Dessas características, o tamanho da ORF e as frequências relativas dos nucleotídeos foram encontradas na literatura [70]. As posições de início e fim das ORFs foram propostas neste trabalho. Abaixo são listados os tipos de características utilizados para criação do modelo de classificação:

1. Inteiros:
   (a) Posição de início da primeira ORF;
   (b) Posição do fim da primeira ORF;
   (c) Posição de início da menor ORF;
   (d) Posição do fim da menor ORF;

(e) Posição de início da maior ORF;

(f) Posição do fim da maior ORF.

2. Reais:

(a) Frequências relativas dos di, tri e tetra-nucleotídeos, por exemplo, 'AA', 'CAC', 'TGAG';

(b) Tamanho relativo da primeira ORF sobre o tamanho do transcrito;

(c) Tamanho relativo da maior ORF sobre o tamanho do transcrito;

(d) Tamanho relativo da menor ORF sobre o tamanho do transcrito.

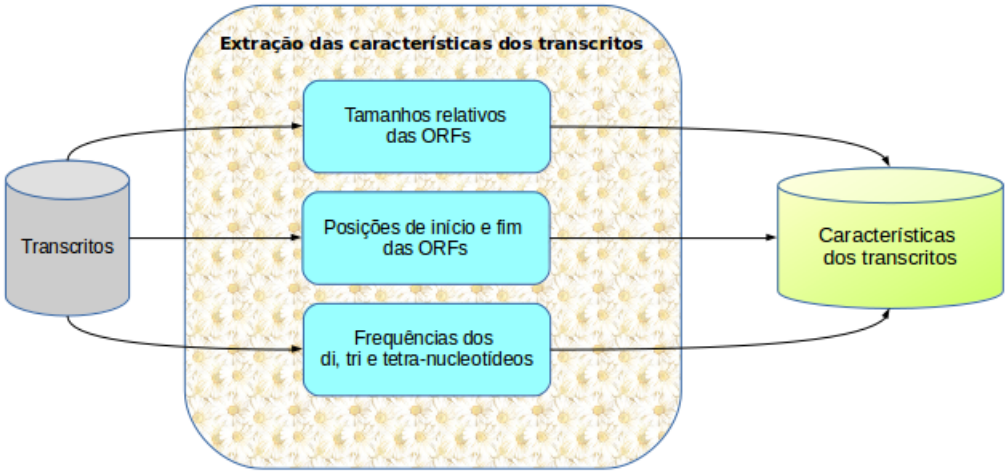


Figura 4.2: Extração das características dos transcritos.

A Figura 4.2 representa o método de extração das características dos transcritos que foram utilizados como dados de entrada do *Random Forest*. Arquivos FASTA [27] são obtidos do banco de transcritos, dos quais são extraídas suas características.

Características das ORFs

Para obter às características relacionadas as ORFs dos transcritos, como listado na Seção 4.1.1, um *script* em *Perl* [63] foi utilizado. As ORFs e suas posições de início e fim foram extraídas dos transcritos seguindo o modelo de extração de ORFs do *NCBI ORF Finder*[58]:

- A sequência do transcrito é lida no sentido 5' ao 3';
- O códon de início da ORF é a sequência "ATG";
- O tamanho da ORF deve ser maior que 30 bps;
- O fim da ORF é obtido com a leitura dos *stop codons* "TAA", "TAG"e "TGA"ou até que o fim do transcrito seja alcançado.

Frequências relativas dos di, tri e tetra-nucleotídeos

As frequências relativas dos di, tri e tetra-nucleotídeos dos transcritos, como listado na Seção 4.1.1, foram as mesmas obtidas pelo Schneider [70]. Foi calculada a frequência relativa média de todos os di, tri e tetra-nucleotídeos nos N possíveis arranjos dos nucleotídeos nos transcritos [70]. Essa média foi calculada por não se saber exatamente onde é iniciada a leitura do transcrito. A frequência relativa média dos transcritos foi obtida seguindo os passos abaixo:

1. Os N possíveis arranjos dos nucleotídeos em um transcrito são selecionados;
2. Para cada possível arranjo é calculado a frequência relativa do nucleotídeo dividindo a frequência obtida pelo tamanho do transcrito dividido pelo o tamanho da sequência dos nucleotídeos;
3. É gerado uma média de todas as frequências relativas encontradas.

4.2 Testes

As características foram divididas em três conjuntos, referentes aos tamanhos relativos das ORFs, suas posições de início e fim e as frequências dos di, tri e tetra-nucleotídeos. Sendo assim, o experimento foi dividido em seis testes para análise das características:

1. Tamanho relativo das ORFs + Posições de início e fim das ORFs;
2. Tamanho relativo das ORFs;
3. Posições de início e fim das ORFs;
4. Frequências dos nucleotídeos;
5. Tamanho relativo das ORFs + Frequência dos nucleotídeos;
6. Tamanho relativo das ORFs + Posições de início e fim das ORFs + Frequência dos nucleotídeos.

4.2.1 Organização dos Testes

Para avaliar a qualidade do modelo preditivo criado, o conjunto de dados selecionados para os testes foi dividido em três grupos. Para todos os testes, foram utilizados dados contendo 80% dos dados para treinamento do modelo e 20% para teste.

Os testes foram realizados com os seguintes dados:

1. Dados balanceados, com PCTs selecionadas aleatoriamente. Conjunto de transcritos com a mesma quantidade de lncRNAs e PCTs;
2. Dados balanceados, com PCTs selecionadas por método de clusterização. Conjunto de transcritos com a mesma quantidade de lncRNAs e PCTs;
3. Dados desbalanceados. Conjunto de transcritos com a quantidade de PCTs superior a de lncRNAs.

4.2.2 Validação das importâncias das características

Para avaliar a importância das características que foram obtidas utilizando o *Random Forest*, um conjunto de transcritos com a mesma quantidade de lncRNAs e PCTs, com PCTs selecionadas por método de clusterização (item 2 da Seção 4.2.1), foi utilizado para criar um modelo preditivo com dois diferentes conjuntos de características.

No primeiro, os di, tri e tetra-nucleotídeos mais importantes foram utilizados como as únicas características do modelo. No segundo, todas as características mais importantes foram utilizadas na construção do modelo.

4.3 Detalhes da Implementação

Nesta Seção, detalhes de como o projeto foi implementado serão apresentados. Na Seção 4.3.1, dados, bibliotecas utilizadas e parâmetros importantes para a construção de um bom modelo preditivo utilizando *Random Forest* e SVM são descritos.

4.3.1 Criação do Modelo de Classificação *Random Forest*

Para a criação de um modelo preditivo utilizando os algoritmos *Random Forest* e SVM foram utilizadas respectivamente as bibliotecas *sklearn.ensemble* e *sklearn.svm* do *Python* [65]. A biblioteca *sklearn.ensemble* disponibiliza uma série de métodos *ensemble* de classificação. Dentre esses métodos, foi selecionado o *RandomForestClassifier* que é um classificador que utiliza o algoritmo *Random Forest*. A biblioteca *sklearn.svm* disponibiliza o método SVC (*Support Vector Classification*) que utiliza o algoritmo SVM para classificação de dados.

Dados utilizados para treinamento e teste dos modelos classificadores

Como apresentado na Seção 4.2.1 o conjunto de dados selecionados para os testes foi dividido em três grupos. Para os três grupos foi gerado um arquivo *Comma Separated Values* (CSV) [23] contendo todas as características mencionadas na Seção 4.1.1 para serem utilizados como os dados de treinamento e teste dos modelos de classificação.

A quantidade de dados utilizados nos modelos levou em consideração o tempo gasto para sua construção. Com isso, apenas parte de todos so dados disponíveis foram selecionados de forma a obter um melhor desempenho quanto ao tempo de construção dos modelos.

Todos os tipos de dados utilizados para a construção do modelo de classificação apresentam a mesma quantidade de lncRNAs para treinamento e teste, 20.000 e 5.000 respectivamente. Para os dados balanceados com PCTs selecioandas aleatoriamente foram selecionadas 20.000 PCTs para treinamento e 5.000 PCTs para teste. Para os dados balanceados com PCTs selecionadas por método de clusterização foi utilizado o Clustal Omega [24] que gerou 226 diferentes clusters. Cada cluster contribuiu para serem selecionadas 20.000 PCTs para treinamento e 5.000 para teste. Para os dados desbalanceados um total de 75.200 PCTs foram utilizadas para treinamento e 18.800 para testes.

Opções de Treinamento do *Random Forest*

Para a criação de um modelo *Random Forest* com boa *performance*, alguns parâmetros que podem ser ajustadas para melhorar o poder preditivo do modelo foram selecionados:

- **max_features**: Determina o número máximo de características utilizadas pelo Random-Forest na montagem de uma árvore individual. Há várias opções disponíveis em *Python* para atribuir o número máximo de características. Algumas delas são:

  1. Auto/None: Simplesmente não ocorre nenhuma restrição em uma árvore individual. Faz com que o Random-Forest utilize todas as variáveis que julgar a melhor para a montagem de cada árvore;
  2. Sqrt: Essa opção terá a raiz quadrada do número total de variáveis em uma execução individual. Por exemplo, se o número total de variáveis for 100, só podemos pegar 10 delas em uma árvore individual. "Log2"é outro tipo de opção semelhante para *max_features*;
  3. *float values* (Ex: 0.2): Esta opção permite que o *Random Forest* tome 20% das variáveis em execução individual. Podemos atribuir e valorizar em um formato "0.x"onde queremos que x% dos recursos sejam considerados;

  Aumentar *max_features* geralmente melhora o desempenho do modelo já que em cada nó temos um maior número de opções a serem consideradas. No entanto, isso não é necessariamente verdade, pois isso diminui a diversidade de cada árvore individual, que é um dos objetivos do Random-Forest. Quanto maior o *max_features* mais lento é o algoritmo;

- **n_estimators**: Número de árvores que se deseja construir em uma floresta. Quanto maior o número de árvores, melhor o desempenho, pois tornam as previsões mais estáveis, mas torna o seu código mais lento;

- **min_sample_leaf**: O número mínimo de amostras necessárias para estar em um nó folha. Um número mínimo de amostras menor torna o modelo mais propenso a capturar ruído nos dados de treinamento;

- **random_state**: A semente usada pelo gerador de números aleatórios. Este parâmetro torna uma solução fácil de replicar. Um valor definido de *random_state* sempre produzirá os mesmos resultados, se for dado com os mesmos parâmetros e dados de treinamento. Se não for utilizado, o gerador de números aleatórios é a instância *RandomState* usada *np.random*;

- **oob_score**: Este é um método aleatório de validação cruzada do *Random Forest*. Este método simplesmente marca todas as observações utilizadas em diferentes árvores e então descobre uma pontuação máxima de votos para cada observação com base em apenas árvores que não usaram esta observação, em particular em seu treinamento.

Opções de treinamento do SVM

Para a criação de um modelo SVM com boa *performance*, alguns parâmetros que podem ser ajustadas para melhorar o poder preditivo do modelo foram selecionados:

- **kernel**: Existem várias opções disponíveis para o kernel como, *linear*, *rbf*, *poly* entre outros (o valor padrão é *rbf*). O Kernel *linear* cria um hiper-plano linear enquanto os *rbf* e *poly* são úteis para hiperplanos não-lineares;
- **gamma**: É o coeficiente para os Kernels não-lineares (*rbf*, *poly*). Quanto maior o valor do gamma, mais o modelo tentará ajustar aos dados de treinamento, isto é, ocorre um erro da generalização, o que pode causar o problema do ajuste excessivo (*overfitting*);
- **C**: O parâmetro C indica ao SVM o quanto se deseja evitar classificar incorretamente cada exemplo de treinamento. Para valores grandes de C, a otimização irá escolher um hiperplano de menor margem, se esse hiperplano fizer um trabalho melhor ao obter todos os pontos de treinamento classificados corretamente. Por outro lado, um valor muito pequeno de C fará com que o otimizador procure uma margem maior separando o hiperplano, mesmo se esse hiperplano classifica incorretamente mais pontos. Para valores muito pequenos de C, deve-se obter exemplos mal classificados, muitas vezes, mesmo se seus dados de treinamento são linearmente separáveis;
- **random_state**: A semente usada pelo gerador de números aleatórios. Este parâmetro torna uma solução fácil de replicar. Um valor definido de *random_state* sempre produzirá os mesmos resultados se for dado com os mesmos parâmetros e dados de treinamento.

Detalhes da Máquina

A máquina utilizada foi um Ultrabook da Samsung com processador Intel i7, com 8G de memória RAM. O sistema operacional utilizado foi o Ubuntu 16.04.1 LTS.

Capítulo 5

Resultados

Neste capítulo, serão discutidos os resultados obtidos a partir do método descrito no capítulo anterior. Na Seção 5.1, serão apresentados as medidas utilizadas para medir a *performance* do método implementado. Na Seção 5.2 os resultados de cada teste listado na Seção 4.2 serão analisados. Na Seção 5.3 as características mais importantes para classificação de lncRNAs são extraídas. Na Seção 5.4 serão realizadas observações gerais sobre os experimentos, com a comparação dos resultados com outros obtidos na literatura. Por fim, na Seção 5.5 é proposto um modelo preditivo utilizando as características mais importantes obtidas.

Neste trabalho, foram utilizados dois bancos de dados: Ensembl [26] e HAVANA [28]. Esses bancos de dados foram utilizados para fornecer a base de dados necessária para a construção do modelo classificador. PCTs com uma sequência maior que 200 bases nitrogenadas foram usados como dados de treinamento negativos e lncRNAs como dados de treinamento positivos. O genoma selecionado foi o GRCh38 do *Homo sapiens* (humano) [26, 28].

5.1 Desempenho

Foram escolhidas algumas medidas estatísticas para avaliar a *performance* do modelo *Random Forest*. Essas medidas buscam observar se o modelo está se comportando da maneira esperada, isso é, avaliar se o sistema está retornando os valores esperados par os dados de teste. Para cada caso de teste apresentado na Seção 4.2 foi construído uma matriz de confusão [11]. Cada matriz apresenta um conjunto de quatro informações importantes para a análise do modelo:

- **Verdadeiro positivo (VP)**: O número de dados corretamente identificados como lncRNAs;
- **Falso positivo (FP)**: O número de dados incorretamente identificados como lncRNAs;
- **Verdadeiro negativo (VN)**: O número de dados corretamente identificados como PCTs;
- **Falso negativo (FN)**: O número de dados incorretamente identificados como PCTs.

Utilizando os dados das matrizes de confusão, para análise da *performance* do modelo, as seguintes medidas foram extraídas:

- **Acurácia**: É a capacidade de diferenciar os dados de lncRNAs e PCTs corretamente. Matematicamente, pode ser calculado como:

$$\frac{VP + VN}{VP + VN + FP + FN}$$

- **Precisão**: Determina quantos dados que foram classificados como lncRNA são relevantes. Matematicamente, pode ser calculado como:

$$\frac{VP}{VP + FP}$$

- **Sensibilidade**: É a capacidade de classificar os dados como lncRNA corretamente. Matematicamente, pode ser calculada como:

$$\frac{VP}{VP + FN}$$

- **Especificidade**: É a capacidade de classificar os dados como PCTs corretamente. Matematicamente, pode ser calculada como:

$$\frac{VN}{VN + FP}$$

- ***F-measure***: É uma medida da precisão para um teste. As duas medidas precisão e sensibilidade são usadas juntas para fornecer uma única medição para um sistema. Matematicamente, pode ser calculada como:

$$\frac{2VP}{2VP + FP + FN}$$

A pontuação OOB, que é a pontuação do conjunto de dados de treinamento obtido usando uma estimativa *out-of-bag*, também foi adicionada à Tabela de *performance* para análise.

5.2 *Performance* dos Testes

Nesta Seção os resultados dos testes listados na Seção 4.2 serão apresentados. Para cada teste, os dados foram selecionados de três diferentes formas, balanceados com PCTs selecionadas aleatoriamente e por método de clusterização, além de dados desbalanceados apresentando mais PCTs, como descrito na Seção 4.2.1.

Por questões de desempenho, para que o tempo da construção do modelo não seja elevado, apenas parte dos dados disponíveis foi utilizado. Para os dados balanceados,

foram usados 20.000 lncRNAs e 20.000 PCTs para treinamento. Para teste, foram utilizados 5.000 lncRNAs e 5.000 PCTs. Para os dados desbalanceados, foram usados 20.000 lncRNAs e 75.200 PCTs para treinamento. Para teste, foram utilizados 5.000 lncRNAs e 18.800 PCTs.

5.2.1 Teste 1: Tamanho das ORFs e Posições das ORFs

Nesta Seção, o primeiro teste da Seção 4.2 é analisado. Esse teste consiste em analisar a *performance* do *Random Forest* utilizando as características do tamanho relativo das ORFs e suas posições de início e de fim.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados, com PCTs selecionadas aleatoriamente. A Tabela 5.1 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.2 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.1: Teste 1 para dados balanceados com PCTs selecionadas aleatoriamente.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4751	249
lncRNA	79	4921
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4719	281
lncRNA	214	4786

Tabela 5.2: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 1	97	95	98	95
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 1	95	94	96	94

É possível perceber pela Tabela 5.2 que o modelo *Random Forest* apresenta uma *performance* melhor, quando comparado ao SVM. O modelo mostra por sua sensibilidade, que possui boa capacidade de classificar os dados como lncRNA corretamente. O modelo também apresenta um bom comportamento ao classificar os dados como PCTs corretamente, como mostrado em sua especificidade.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados, com PCTs selecionadas por método de clusterização [24]. A Tabela 5.3 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.4 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.3: Teste 1 para dados com PCTs selecionadas por método de clusterização.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4961	39
lncRNA	45	4955
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4931	69
lncRNA	246	4754

Tabela 5.4: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 1	99	99	99	99
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 1	97	99	95	99

É possível perceber pela Tabela 5.4 que o modelo *Random Forest* apresenta uma *performance* melhor quando comparado ao SVM. Os resultados obtidos foram melhores, quando comparados aos obtidos com PCTs selecionadas aleatoriamente. Isso era esperado visto que, com dados clusterizados, é possível obter uma melhor generalização do modelo.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, dados que contêm mais PCTs. A Tabela 5.5 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.6 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.5: Teste 1 com dados desbalanceados, apresentando mais PCTs.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	18285	515
lncRNA	246	4754
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	18229	571
lncRNA	1204	3796

Tabela 5.6: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*	
Conjunto	F-measure
Teste 1	92
Performance do modelo SVM	
Conjunto	F-measure
Teste 1	81

É possível perceber, na Tabela 5.6 uma diferença relevante do *F-measure*, para os modelos *Random Forest* e SVM. Isso demonstra que, para o Teste 1 com dados desbalanceados, o *Random Forest* teve um melhor poder preditivo que o SVM.

5.2.2 Teste 2: Tamanho das ORFs

Nesta Seção, o segundo teste da Seção 4.2 é analisado. Esse teste consiste em analisar a *performance* do *Random Forest* utilizando as características de tamanho relativo das ORFs apenas.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. A Tabela 5.7 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.8 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.7: Teste 2 para dados balanceados com PCTs selecionadas aleatoriamente.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4755	245
lncRNA	79	4921
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4780	220
lncRNA	131	4869

Tabela 5.8: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 2	96	95	98	95
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 2	96	96	97	96

Os valores obtidos na Tabela 5.8 mostram que a *performance* do *Random Forest* e SVM foram muito semelhantes, com a mesma acurácia quando aplicadas ao Teste 2. Ambos os modelos mostram por sua sensibilidade que possuem boa capacidade de classificar os dados como lncRNAs corretamente. Os modelos também apresentam um bom comportamento ao classificar os dados como PCTs corretamente, como mostrado em sua especificidade. Isso comprova a importância do tamanho relativo das ORFs para um modelo de classificação de lncRNAs.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas por método de clusterização [24]. A Tabela 5.9 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.10 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.9: Teste 2 com dados com PCTs selecionadas por método de clusterização.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4763	237
lncRNA	79	4921
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4797	203
lncRNA	123	4877

Tabela 5.10: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 2	97	95	98	95
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 2	97	96	98	96

Os valores obtidos na Tabela 5.10 mostram uma *performance* um pouco melhor que aquelas apresentadas na Tabela 5.8 devido a clusterização das PCTs, mas ainda assim, o *Random Forest* e SVM apresentaram *performances* muito semelhantes.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, que apresentam mais PCTs. A Tabela 5.11 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.12 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.11: Teste 2 com dados desbalanceados apresentando mais PCTs.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	18288	512
lncRNA	282	4718
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	18278	522
lncRNA	287	4713

Tabela 5.12: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*	
Conjunto	F-measure
Teste 2	92
Performance do modelo SVM	
Conjunto	F-measure
Teste 2	92

Os valores obtidos para o *F-measure* da Tabela 5.12 comprovam o a similaridade entre os modelos *Random Forest* e SVM quando utilizados com as características de tamanho relativo das ORFs. Para esse teste, não houve impacto na *performance* dos modelos quando os dados são desbalanceados ou não. Isso comprova que as características de tamanho relativo das ORFs elevam a qualidade dos modelos preditivos.

5.2.3 Teste 3: Posições das ORFs

Nesta Seção, o terceiro teste da Seção 4.2 é analisado. Esse teste consiste em analisar a *performance* do *Random Forest* utilizando as características das posições de início e fim das ORFs apenas.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. A Tabela 5.13 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.14 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.13: Teste 3 com dados balanceados com PCTs selecionadas aleatoriamente.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4558	442
lncRNA	91	4909
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4257	743
lncRNA	69	4931

Tabela 5.14: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 3	95	92	98	91
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 3	92	87	99	85

É possível perceber que a *performance* do *Random Forest* foi superior a do SVM. Observa-se que o SVM teve problemas ao classificar PCTs corretamente, como mostra sua especificidade. A utilização das características de posição de início e fim das ORFs geraram um modelo classificador de bom desempenho, quando o algoritmo *Random Forest* é utilizado.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas por método de clusterização [24]. A Tabela 5.15 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.16 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.15: Teste 3 com dados com PCTs selecionadas por método de clusterização.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4539	461
lncRNA	91	4909
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	3915	1085
lncRNA	71	4929

Tabela 5.16: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 3	94	91	98	91
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 3	88	82	99	78

É possível perceber que a *performance* do *Random Forest* manteve-se superior a do SVM, sem grandes diferenças no caso em que os dados possuem PCTs selecionadas por método de clusterização. Observa-se que o SVM teve sua acurácia ligada a classificação de PCTs como mostra sua especificidade. A utilização das características de posição de início e fim das ORFs geraram um modelo classificador de bom desempenho, quando utilizando o algoritmo *Random Forest*.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, dados apresentando mais PCTs. A Tabela 5.17 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.18 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.17: Teste 3 com dados desbalanceados apresentando mais PCTs.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	18030	770
lncRNA	376	4624
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	18510	290
lncRNA	2007	2993

Tabela 5.18: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*	
Conjunto	F-measure
Teste 3	88
Performance do modelo SVM	
Conjunto	F-measure
Teste 3	72

Para os dados desbalanceados a *performance* do *Random Forest* foi ainda melhor quando comparado ao SVM, mas inferior àqueles em que os dados utilizados para treinamento e teste do modelo *Random Forest* eram balanceados. A utilização das características de posição de início e fim das ORFs geraram um modelo classificador de médio desempenho, quando o algoritmo *Random Forest* com dados desbalanceados é utilizado.

5.2.4 Teste 4: Frequências dos di, tri e tetra-nucleotídeos

Nesta Seção, o quarto teste da Seção 4.2 é analisado. Esse teste consiste em analisar a *performance* do *Random Forest* utilizando as características das frequências relativas dos di, tri e tetra-nucleotídeos.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. A Tabela 5.19 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.20 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.19: Teste 4 com dados balanceados com PCTs selecionadas aleatoriamente.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4079	921
lncRNA	583	4417
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4381	619
lncRNA	615	4385

Tabela 5.20: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 4	85	83	88	82
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 4	88	88	88	88

Para esse teste a *performance* do SVM foi superior à do *Random Forest*. O *Random Forest* apresentou problemas para definir PCTs corretamente, como é apontado pelo valor de sua especificidade. A utilização das frequências dos di, tri e tetra-nucleotídeos como as únicas características presentes na construção de um modelo preditivo utilizando o *Random Forest* apresentaram um desempenho médio.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados, em que as PCTs foram selecionadas por método de clusterização [24]. A Tabela 5.21 apresenta

uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.22 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.21: Teste 4 com dados as PCTs selecionadas por método de clusterização.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4078	922
lncRNA	591	4409
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4444	556
lncRNA	618	4382

Tabela 5.22: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 4	85	83	88	82
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 4	88	89	88	89

Para esse teste, a *performance* do SVM continou superior à do *Random Forest*. O *Random Forest* apresentou os mesmos problemas de especificidade de quando as PCTs foram selecionada aleatoriamente. A utilização das frequências dos di, tri e tetra-nucleotídeos como as únicas características presentes na construção de um modelo preditivo utilizando o *Random Forest* apresentaram um desempenho não tão bom.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, que apresentam mais PCTs. A Tabela 5.23 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.24 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.23: Teste 4 com dados desbalanceados apresentando mais PCTs.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	18720	80
lncRNA	3122	1878
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	18358	442
lncRNA	1656	3344

Tabela 5.24: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*	
Conjunto	F-measure
Teste 4	53
Performance do modelo SVM	
Conjunto	F-measure
Teste 4	76

Para esse teste, a *performance* do SVM continou superior à do *Random Forest*. O *Random Forest* obteve um F-measure de 53% enquanto a SVM obteve um F-measure de 76% . Isso mostra que esse teste não retorna bons resultados quando os dados são desbalanceados. A utilização das frequências dos di, tri e tetra-nucleotídeos como as únicas características presentes na construção de um modelo preditivo utilizando o *Random Forest* apresentaram um baixo desempenho para dados desbalanceados.

5.2.5 Teste 5: Tamanho das ORFs e Frequências dos di, tri e tetra-nucleotídeos

Nesta Seção, o quinto teste da Seção 4.2 é analisado. Esse teste consiste em analisar a *performance* do *Random Forest*, utilizando as características dos tamanhos relativos das ORFs e as frequências relativas dos di, tri e tetra-nucleotídeos.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. A Tabela 5.25 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.26 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.25: Teste 5 com dados balanceados com PCTs selecionadas aleatoriamente.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4802	198
lncRNA	88	4912
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4815	185
lncRNA	110	4890

Tabela 5.26: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 5	97	96	98	96
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 5	97	96	98	96

A Tabela 5.26 aponta que o *Random Forest* e o SVM apresentaram desempenho semelhante para esse teste. Isso é devido à da presença das características de tamanho relativo das ORFs. Os tamanhos relativos das ORFs elevam a *performance* do modelo preditivo. Assim o *Random Forest*, que não apresentou bom desempenho no teste 4 em que apenas as frequências relativas dos di, tri e tetra-nucleotídeos foram consideradas, passou a obter uma boa performance.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas por método de clusterização [24]. A Tabela 5.27 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.28 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.27: Teste 5 com dados com PCTs selecionadas por método de clusterização.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4824	176
lncRNA	87	4913
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4836	164
lncRNA	105	4895

Tabela 5.28: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 5	97	97	98	96
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 5	97	97	98	97

A Tabela 5.26 aponta que *Random Forest* e a SVM também apresentaram desempenho semelhante quando as PCTs foram clusterizadas. Os tamanhos relativos das ORFs continuaram a elevar a *performance* do modelo preditivo.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, dados apresentando mais PCTs. A Tabela 5.29 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.30 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.29: Teste 5 com dados desbalanceados apresentando mais PCTs.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	18610	190
lncRNA	487	4513
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	18435	365
lncRNA	234	4766

Tabela 5.30: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*	
Conjunto	F-measure
Teste 5	93
Performance do modelo SVM	
Conjunto	F-measure
Teste 5	94

Para os dados desbalanceados, a *performance* do SVM foi um pouco melhor que a do *Random Forest*, mas ainda assim inferior às *performances* com dados balanceados. Os tamanhos relativos das ORFs elevam a *performance* do modelo preditivo. Assim o *Random Forest*, que não apresentou bom desempenho no teste 4 em que apenas as frequências relativas dos di, tri e tetra-nucleotídeos foram consideradas, passou a obter uma boa performance.

5.2.6 Teste 6: Tamanho das ORFs, Posições das ORFs e Frequências dos di, tri e tetra-nucleotídeos

Nesta Seção, o sexto e último teste da Seção 4.2 é analisado. Esse teste consiste em analisar a *performance* do *Random Forest* utilizando as características dos tamanhos relativos das ORFs, suas posições de início e fim além das frequências relativas dos di, tri e tetra-nucleotídeos.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. A Tabela 5.31 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.32 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.31: Teste 6 com dados balanceados com PCTs selecionadas aleatoriamente.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4698	302
lncRNA	55	4945
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4858	142
lncRNA	509	4491

Tabela 5.32: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 6	96	94	99	94
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 6	93	97	90	97

Para um conjunto de dados contendo todas as características, é possível perceber que o *Random Forest* apresentou uma melhor acurácia. O *Random Forest* também apresentou uma sensibilidade de 99%, indicando que o modelo funcionou muito bem para classificar lncRNAs corretamente. O bom desempenho do modelo classificador deve-se as características de tamanho relativo das ORFs e suas posições de início e fim, que elevam a *performance* do modelo preditivo.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados, com PCTs selecionadas por método de clusterização [24]. A Tabela 5.33 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.34 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.33: Teste 6 com dados com PCTs selecionadas por método de clusterização.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4955	45
lncRNA	46	4954
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4326	674
lncRNA	65	4935

Tabela 5.34: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 6	99	99	99	99
Performance do modelo *SVM*				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 6	93	88	99	87

Para um conjunto de dados contendo PCTs selecionadas por método de clusterização o modelo teve uma ótima performance. Isso é esperado pois a clusterização traz ao modelo uma maior generalização, elevando o seu poder preditivo. O bom desempenho do modelo classificador deve-se à clusterização das PCTs e às características de tamanho relativo das ORFs e suas posições de início e fim que elevam a *performance* do modelo preditivo.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, apresentando mais PCTs. A Tabela 5.35 apresenta uma comparação entre as matrizes de confusão geradas pelo *Random Forest* e SVM. Por último, a Tabela 5.36 apresenta os valores das medidas estatísticas listadas na Seção 5.1.

Tabela 5.35: Teste 6 com dados desbalanceados apresentando mais PCTs.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	18420	380
lncRNA	266	4734
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	15945	2855
lncRNA	457	4543

Tabela 5.36: *Performance* dos modelos *Random Forest* e SVM.

Performance do modelo *Random Forest*	
Conjunto	F-measure
Teste 6	93
Performance do modelo SVM	
Conjunto	F-measure
Teste 6	73

Para um conjunto de dados desbalanceado, o modelo *Random Forest* teve uma *performance* muito superior ao SVM. O bom desempenho do modelo classificador deve-se às características de tamanho relativo das ORFs e suas posições de início e fim, que elevam a *performance* do modelo preditivo.

5.3 Extração de Características

Nesta Seção os as características mais importantes para a classificação de lncRNAs são extraídas do modelo *Random Forest* construído na Seção 5.2 para cada um dos testes listados na Seção 4.2.

Para cada teste, os dados foram selecionados de três diferentes formas, balanceados com PCTs selecionadas aleatoriamente e por método de clusterização, além de dados desbalanceados apresentando mais PCTs, como descrito na Seção 4.2.1. Para cada teste é apresentando um Gráfico contendo as características apontadas como mais importantes pelo *Random Forest*, com exceção do teste 4 em que uma tabela com os 60 di, tri e tetra-nucleotídeos mais importantes é apresentada.

5.3.1 Teste 1: Tamanho das ORFs e Posições das ORFs

Nesta Seção, o primeiro teste da Seção 4.2 é analisado. Esse teste consiste em apresentar as características mais importantes apontadas pelo *Random Forest* utilizando como conjunto de características o tamanho relativo das ORFs e suas posições de início e de fim.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados, com PCTs selecionadas aleatoriamente. O gráfico mostrado na Figura 5.1 apresenta a lista das características mais importantes para o teste implementado.

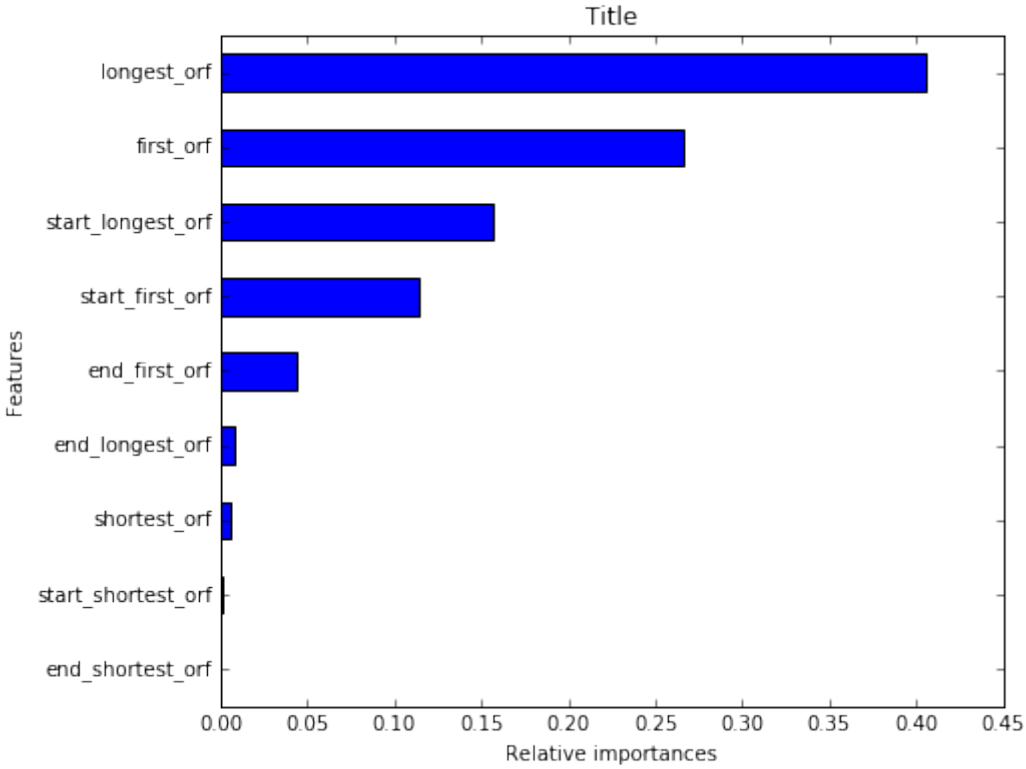


Figura 5.1: Teste 1 (PCTs Aleatórias): Importância relativa das características.

O gráfico da Figura 5.1 confirma o fato, já conhecido na literatura, de que o tamanho da ORF é realmente relevante para a classificação dos lncRNAs [7]. É possivel perceber também que as posições de início da primeira e maior ORF também foram importantes para a classificação do modelo preditivo.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados, com PCTs selecionadas por método de clusterização [24]. O gráfico mostrado na Figura 5.2 apresenta a lista das características mais importantes para o teste implementado.

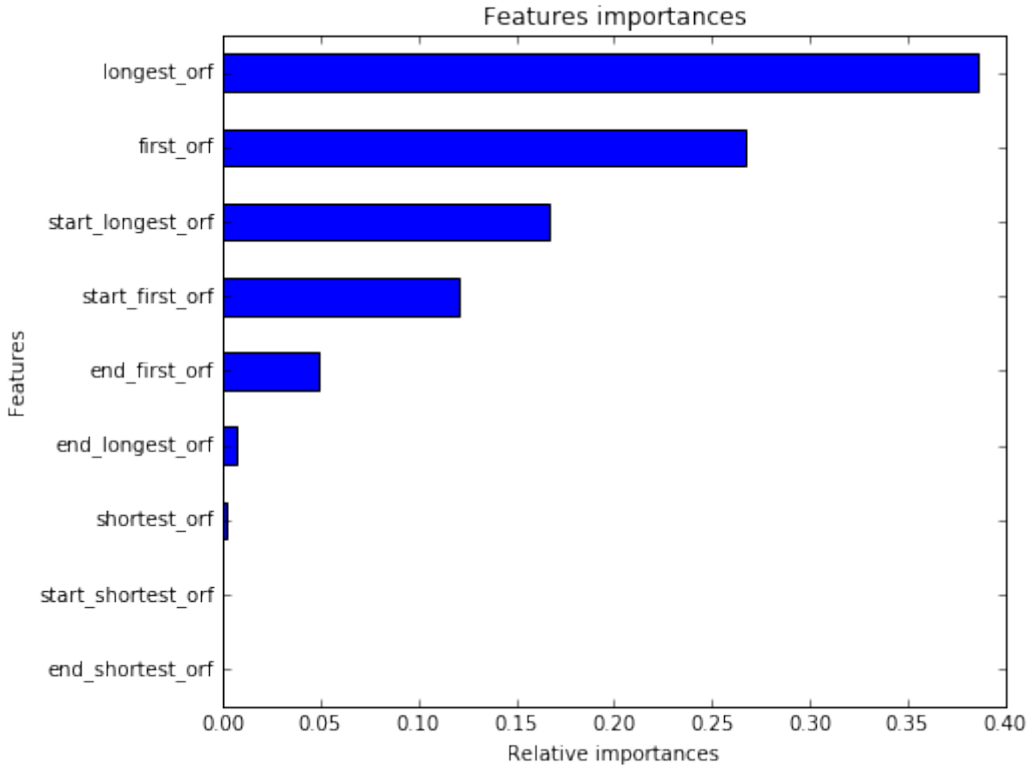


Figura 5.2: Teste 1 (PCTs Clusterizadas): Importância relativa das características.

O gráfico da Figura 5.2 não ficou muito diferente do gráfico 5.1, confirmando que o tamanho da ORF é realmente relevante para a classificação de um lncRNA e que as posições de início da primeira e maior ORF também foram importantes para a classificação do modelo preditivo.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, dados que contêm mais PCTs. O gráfico mostrado na Figura 5.3 apresenta a lista das características mais importantes para o teste implementado.

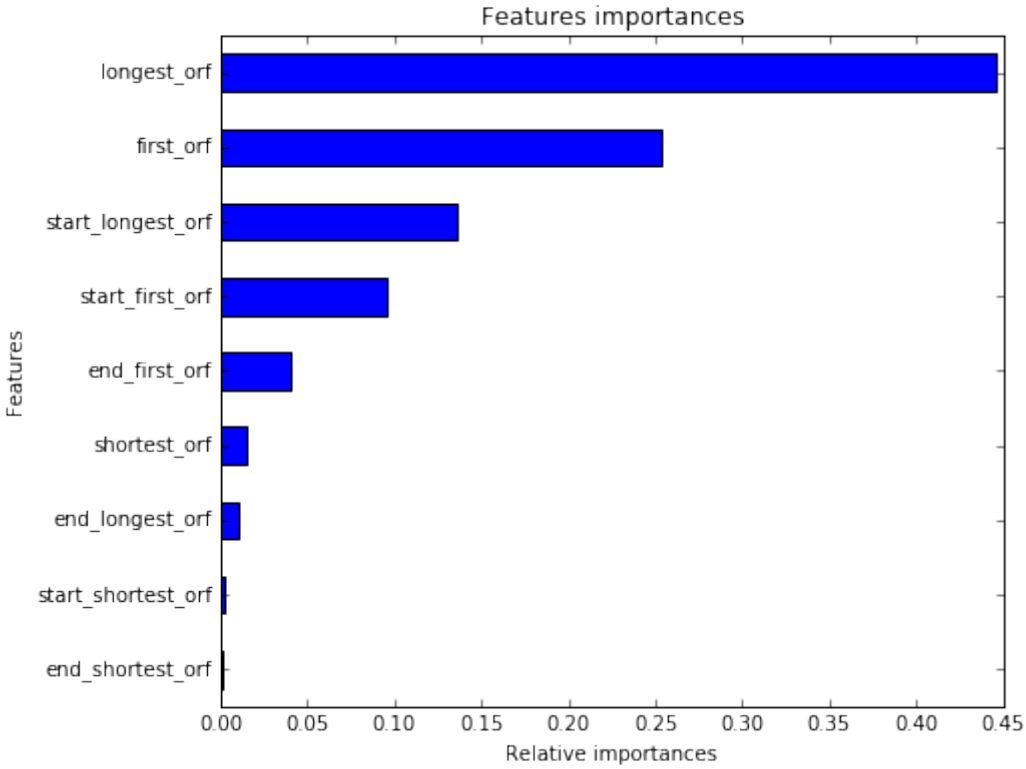


Figura 5.3: Teste 1 (Desbalanceado): Importância relativa das características.

O gráfico da Figura 5.3 é muito semelhante aos gráficos 5.1 e 5.2, apresentando apenas a posição do fim da maior ORF como mais importante que o tamanho relativo da menor ORF. Isso demonstra que as características importantes deste teste sofrem pouca influência de dados balanceados ou não.

5.3.2 Teste 2: Tamanho das ORFs

Nesta Seção, o segundo teste da Seção 4.2 é analisado. Esse teste consiste em apresentar as características mais importantes apontadas pelo *Random Forest* utilizando como conjunto de características o tamanho relativo das ORFs apenas.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. O gráfico mostrado na Figura 5.4 apresenta a lista das características mais importantes para o teste implementado.

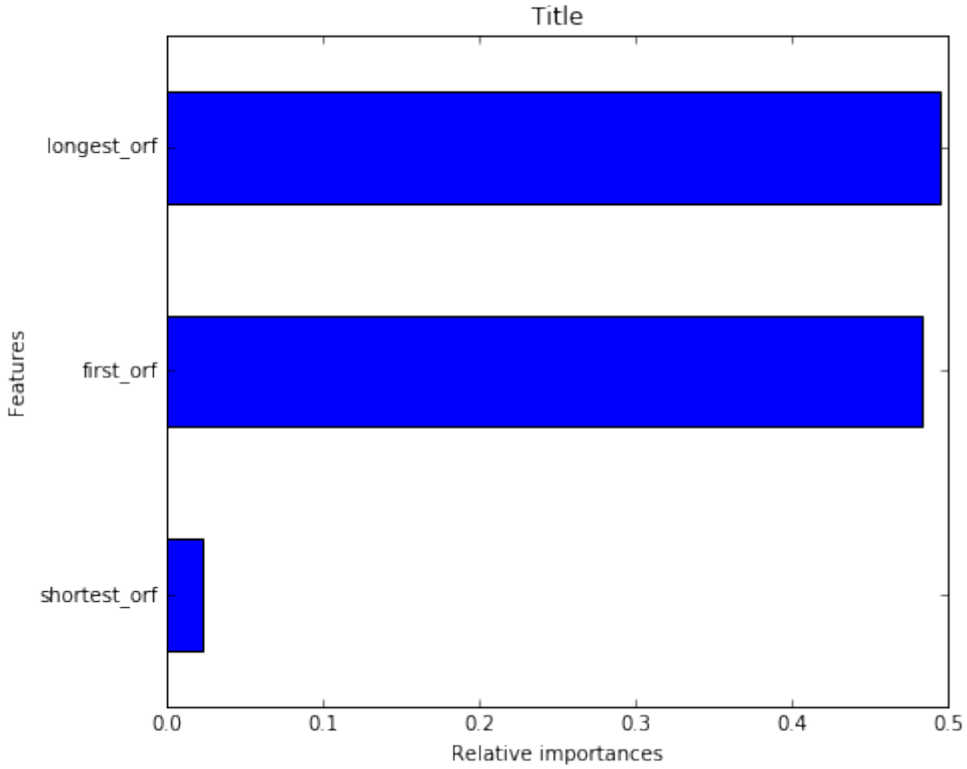


Figura 5.4: Teste 2 (PCTs Aleatórias): Importância relativa das características.

O gráfico da Figura 5.4 comprova o que já foi apontado no primeiro teste. O tamanho relativo da primeira e maior ORF são mais importantes que o tamanho relativo da menor ORF.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas por método de clusterização [24]. O gráfico mostrado na Figura 5.5 apresenta a lista das características mais importantes para o teste implementado.

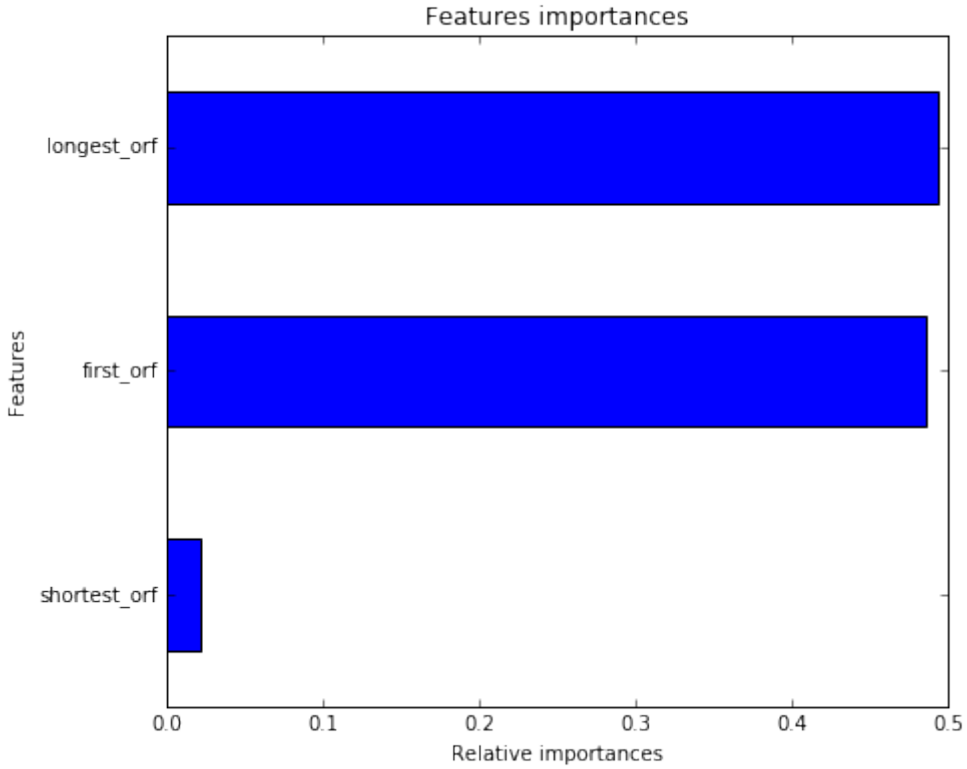


Figura 5.5: Teste 2 (PCTs Clusterizadas): Importância relativa das características.

O gráfico da Figura 5.5 apresentou resultados muito semelhantes aos da Figura 5.4 o que era esperado devido a semelhança das medidas de *performance* apresentadas nas Tabelas 5.8 e 5.10.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, que apresentam mais PCTs. O gráfico mostrado na Figura 5.6 apresenta a lista das características mais importantes para o teste implementado.

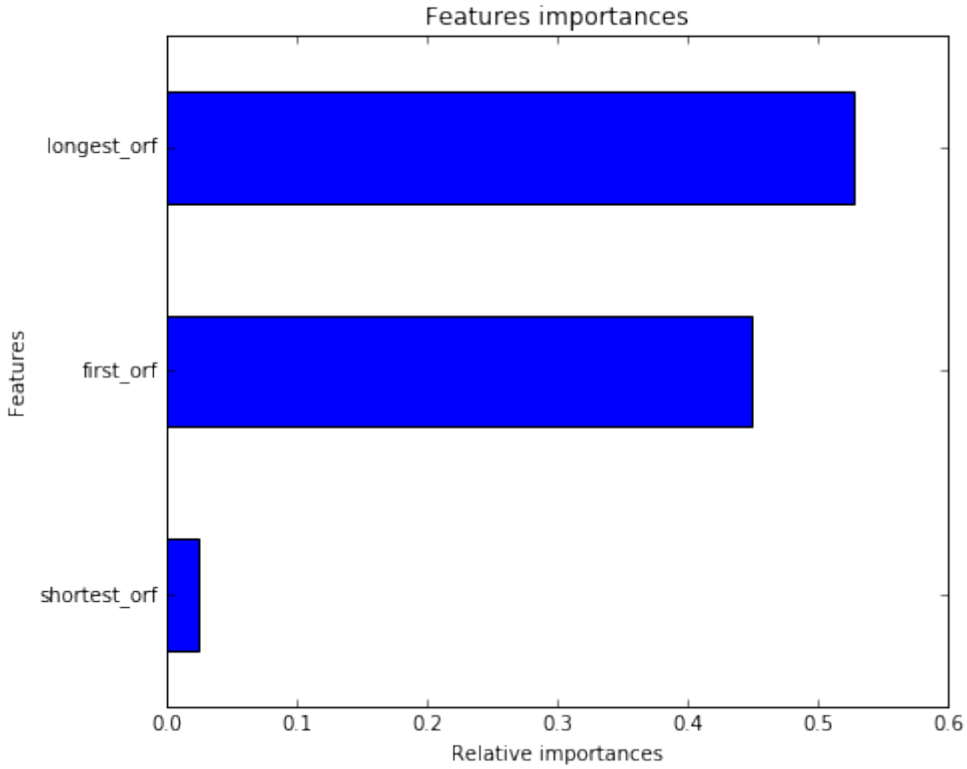


Figura 5.6: Teste 2 (Desbalanceado): Importância relativa das características.

O gráfico da Figura 5.6, como nos gráficos 5.4 e 5.5 também apresentam o tamanho relativo da maior e primeira ORF como mais importantes que o da menor ORF. Confirma-se, então, que a importância das características para esse teste não são influenciadas por dados balanceados ou não.

5.3.3 Teste 3: Posições das ORFs

Nesta Seção, o terceiro teste da Seção 4.2 é analisado. Esse teste consiste em apresentar as características mais importantes apontadas pelo *Random Forest* utilizando como conjunto de características as posições de início e fim das ORFs apenas.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. O gráfico mostrado na Figura 5.7 apresenta a lista das características mais importantes para o teste implementado.

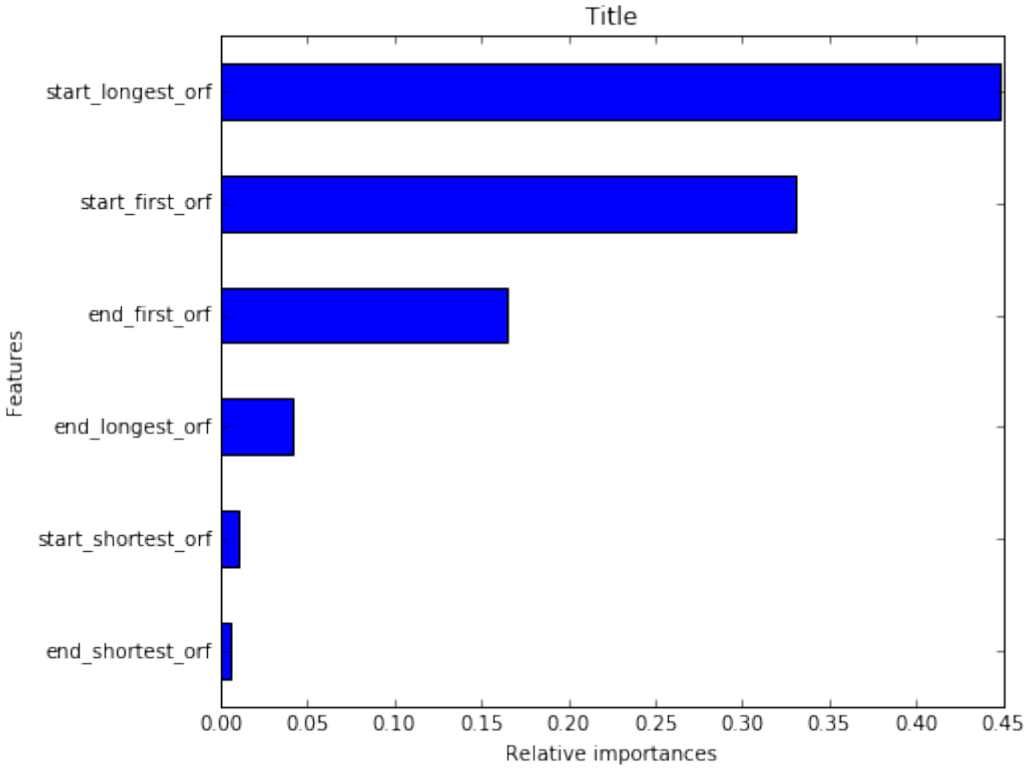


Figura 5.7: Teste 3 (PCTs Aleatórias): Importância relativa das características.

Percebe-se, pelo gráfico da Figura 5.7, que as posições de início da maior e primeira ORF se manteram no topo das posições mais importantes para a classificação de um transcrito como lncRNA. Isso aponta indícios de que as posições de início e fim das ORFs são relevantes para a classificação de lncRNAs.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas por método de clusterização [24]. O gráfico mostrado na Figura 5.8 apresenta a lista das características mais importantes para o teste implementado.

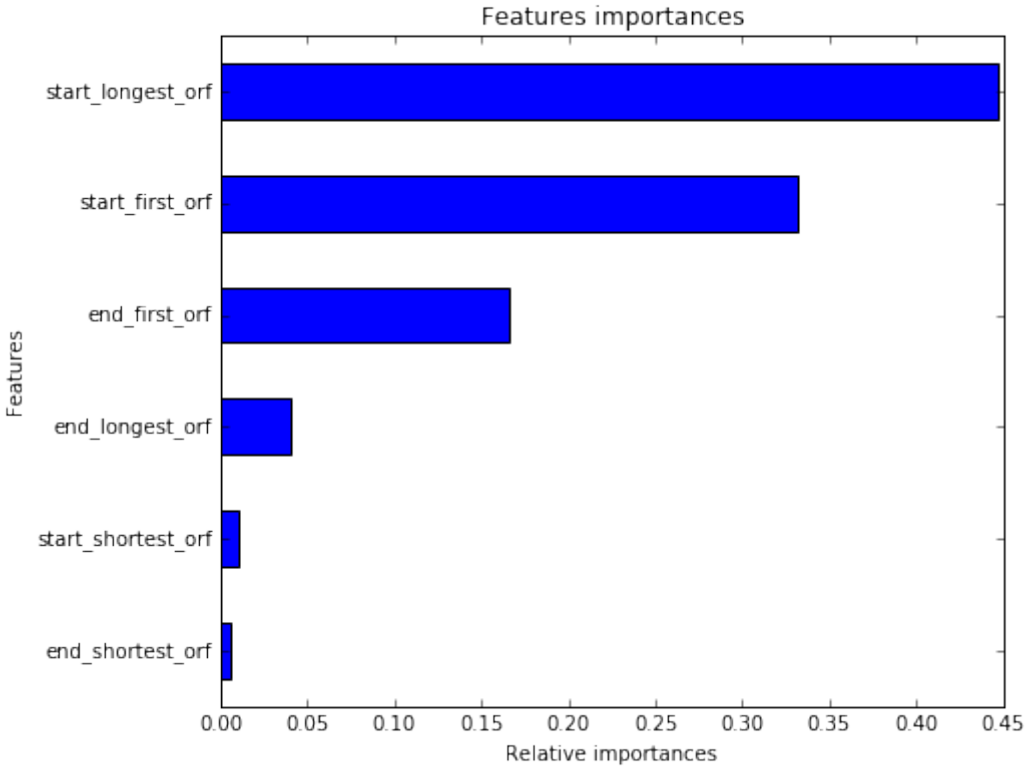


Figura 5.8: Teste 3 (PCTs Clusterizadas): Importância relativa das características.

Percebe-se pelo gráfico da Figura 5.8 que as posições de início da maior e primeira ORF mantiveram-se no topo das posições mais importantes para a classificação de um transcrito como lncRNA, como no gráfico da Figura 5.7. Isso aponta indícios de que as posições de início e fim das ORFs são relevantes para a classificação de lncRNAs.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, dados apresentando mais PCTs. O gráfico mostrado na Figura 5.9 apresenta a lista das características mais importantes para o teste implementado.

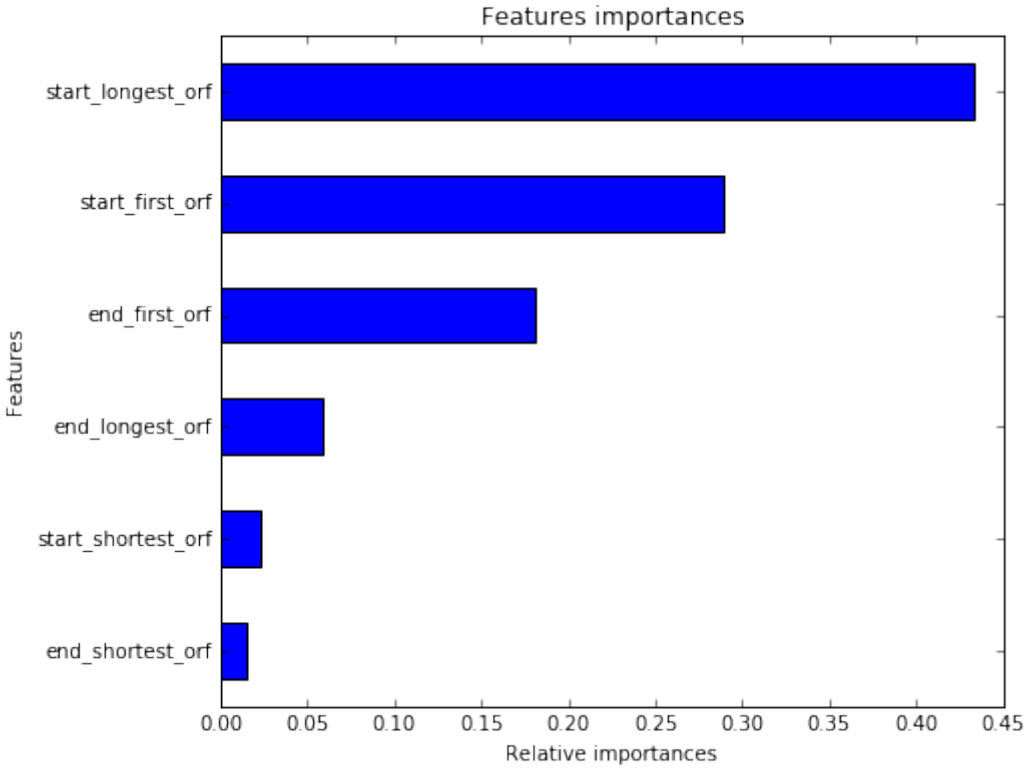


Figura 5.9: Teste 3 (Desbalanceado): Importância relativa das características.

Percebe-se pelo gráfico da Figura 5.9 que as posições de início da maior e primeira ORF mantiveram-se no topo das posições mais importantes para a classificação de um transcrito como lncRNA como nos gráficos das Figuras 5.7 e 5.8. Isso aponta indícios de que as posições de início e fim das ORFs são relevantes para a classificação de lncRNAs.

5.3.4 Teste 4: Frequências dos di, tri e tetra-nucleotídeos

Nesta Seção, o quarto teste da Seção 4.2 é analisado. Esse teste consiste em apresentar as características mais importantes apontadas pelo *Random Forest* utilizando como conjunto de características as frequências relativas dos di, tri e tetra-nucleotídeos.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. A Tabela 5.37 apresenta a lista das características mais importantes para o teste implementado.

Tabela 5.37: Teste 4 (PCTs Aleatórias): 60 frequências mais importantes.

Ranking	Sequências
1-10	tagg, taa, taaa, tag, cga, tcg, ttt, cg, tgga, taat
11-20	atcg, atg, tttt, ga, ttaa, tt, tta, tcga, ctaa, ataa
21-30	ttag, gat, aaa, gtaa, tttc, ctag, gaa, gatg, aata, ct
31-40	taga, tacg, ttta, tct, cgac, cgag, taag, tccc, gaag, acga
41-50	ttcg, cgg, attt, ta, aaat, ctac, aat, aggg, atgg, att
51-60	atga, tgg, ctc, tac, tctc, ccg, aatt, ccga, tgag, gtag

A Tabela 5.19 apresenta a lista dos 60 di, tri e tetra-nucleotídeos mais importantes para a classificação dos lncRNAs. A ausência ou presença desses di, tri e tetra-nucleotídeos no transcrito apresentam indícios de que podem ser relevantes para a classificação de lncRNAs.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados, em que as PCTs foram selecionadas por método de clusterização [24]. A Tabela 5.38 apresenta a lista das características mais importantes para o teste implementado.

Tabela 5.38: Teste 4 (PCTs Clusterizadas): 60 frequências mais importantes.

Ranking	Sequências
1-10	tagg, taa, taaa, tag, cga, ttt, tcg, tttt, tgga, taat
11-20	ttaa, ga, cg, tt, atg, atcg, ctaa, tta, ataa, gtaa
21-30	tcga, ttag, ctag, gatg, aaa, aata, gat, cgac, tacg, taag
31-40	tttc, gaa, ttta, attt, tct, cgag, ttcg, taga, ct, acga
41-50	ta, cgg, ctac, gaag, tgg, aaat, atgg, tccc, aat, att
51-60	atga, tagc, tac, tgag, ctc, aggg, aatt, tctc, ccg, taac

A Tabela 5.21 apresenta a lista dos 60 di, tri e tetra-nucleotídeos mais importantes para a classificação dos lncRNAs. Essa Tabela contém 58 di, tri e tetra-nucleotídeos, presentes na Tabela 5.19 mas rearranjados em ordem diferente. A ausência ou presença desses di, tri e tetra-nucleotídeos no transcrito apresentam indícios de que podem ser relevantes para a classificação de lncRNAs.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, que apresentam mais PCTs. A Tabela 5.39 apresenta a lista das características mais importantes para o teste implementado.

Tabela 5.39: Teste 4 (Desbalanceado): 60 frequências mais importantes.

Ranking	Sequências
1-10	tagg, taa, cga, taaa, tag, tcg, tttt, atg, tgga, cg
11-20	ga, ttt, taat, tttc, ttaa, ctag, gaa, gat, ttag, ataa
21-30	atcg, tcga, gatg, ct, tt, tct, ctaa, tccc, gtaa, aata
31-40	taag, gaag, tta, tgag, atgg, aggg, ctc, taga, tgg, ttta
41-50	aaa, attt, ttcg, atga, acga, cac, tctc, cgac, cgg, tacg
51-60	ta, tac, aaga, agg, agaa, cgag, aggc, ctcc, aaat, tagc

A Tabela 5.39 apresenta a lista dos 60 di, tri e tetra-nucleotídeos mais importantes para a classificação dos lncRNAs. Esses valores apresentam 53 di, tri e tetra-nucleotídeos presentes na Tabela 5.19 e 54 na Tabela 5.21 mas rearranjados em ordem diferente. A ausência ou presença desses di, tri e tetra-nucleotídeos no transcrito apresentam indícios de que podem ser relevantes para a classificação de lncRNAs.

5.3.5 Teste 5: Tamanho das ORFs e Frequências dos di, tri e tetra-nucleotídeos

Nesta Seção, o quinto teste da Seção 4.2 é analisado. Esse teste consiste em apresentar as características mais importantes apontadas pelo *Random Forest* utilizando como conjunto de características os tamanhos relativos das ORFs e as frequências relativas dos di, tri e tetra-nucleotídeos.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. O gráfico mostrado na Figura 5.10 apresenta a lista das características mais importantes para o teste implementado.

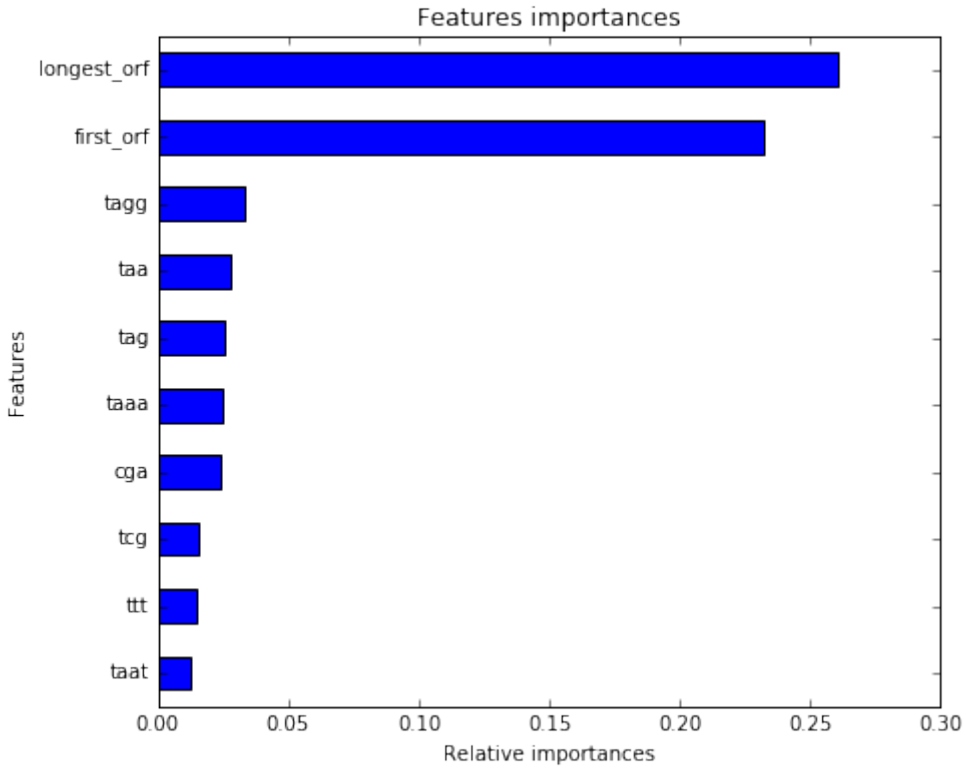


Figura 5.10: Teste 5 (PCTs Aleatórias): Importância relativa das características.

Percebe-se pelo gráfico da Figura 5.10 que o tamanho relativo da maior e primeira ORF são as características mais importantes para a classificação de um transcrito como lncRNA neste teste. As frequências mais importantes presentes no gráfico são as mesmas obtidas na Tabela 5.19 do teste 4 rearranjados em ordem diferente. Isso indica que, além de elevar o desempenho do modelo preditivo, a inclusão dos tamanhos relativos das ORFs pode influenciar nas frequências mais importantes.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas por método de clusterização [24]. O gráfico mostrado na Figura 5.11 apresenta a lista das características mais importantes para o teste implementado.

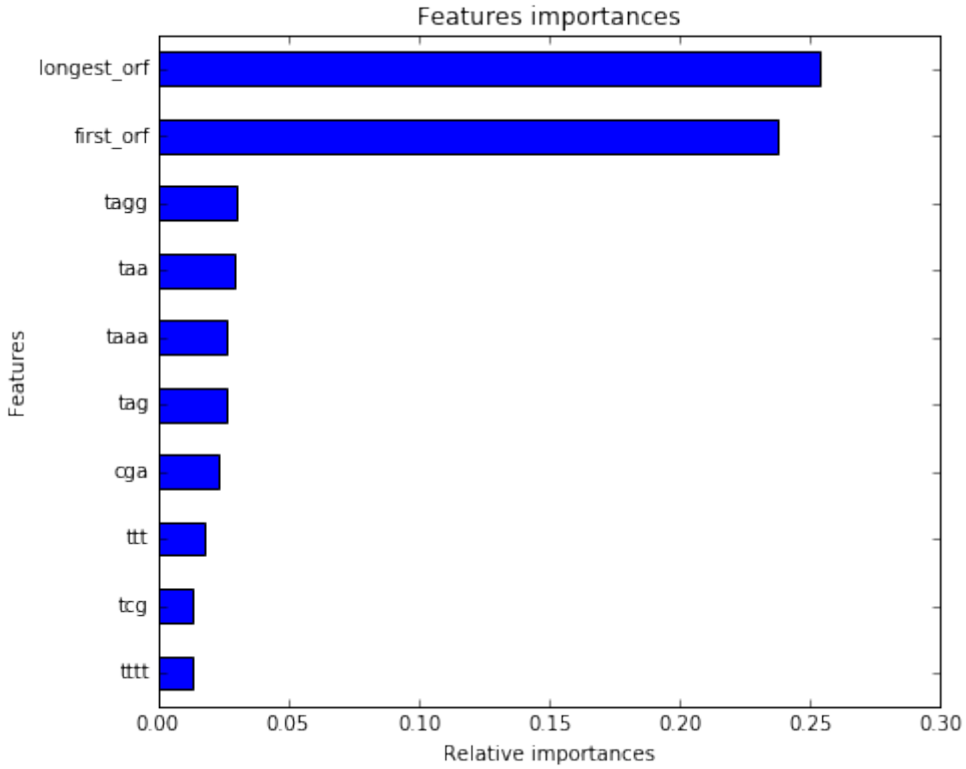


Figura 5.11: Teste 5 (PCTs Clusterizadas): Importância relativa das características.

Percebe-se pelo gráfico da Figura 5.11 que o tamanho relativo da maior e primeira ORF mantiveram-se como as características mais importantes para a classificação de um transcrito como lncRNA. As frequências mais importantes presentes no gráfico são as mesmas obtidas na Tabela 5.21 do teste 4 rearranjados em ordem diferente. Isso indica que, além de elevar o desempenho do modelo preditivo, a inclusão dos tamanhos relativos das ORFs pode influenciar nas frequências mais importantes.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, dados apresentando mais PCTs. O gráfico mostrado na Figura 5.12 apresenta a lista das características mais importantes para o teste implementado.

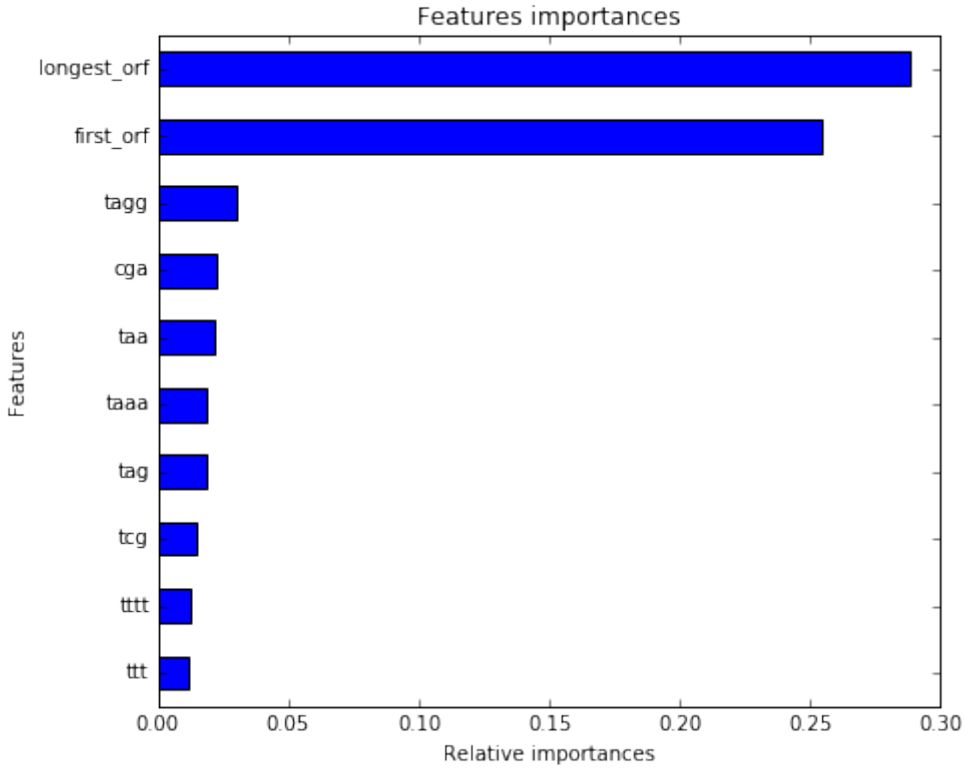


Figura 5.12: Teste 5 (Desbalanceado): Importância relativa das características.

Percebe-se pelo gráfico da Figura 5.12 que o tamanho relativo da maior e primeira ORF mantiveram-se como as características mais importantes para a classificação de um transcrito como lncRNA. As frequências mais importantes presentes no gráfico são as mesmas obtidas na Tabela 5.39 do teste 4 rearranjados em ordem diferente. Isso indica que, apesar de elevar o desempenho do modelo preditivo, a inclusão dos tamanhos relativos das ORFs pode influenciar nas frequências mais importantes, mesmo para dados desbalanceados.

5.3.6 Teste 6: Tamanho das ORFs, Posições das ORFs e Frequências dos di, tri e tetra-nucleotídeos

Nesta Seção, o sexto e último teste da Seção 4.2 é analisado. Esse teste consiste em apresentar as características mais importantes apontadas pelo *Random Forest* utilizando como conjunto de características os tamanhos relativos das ORFs, suas posições de início e fim além das frequências relativas dos di, tri e tetra-nucleotídeos.

Dados balanceados: PCTs selecionadas aleatoriamente

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados com PCTs selecionadas aleatoriamente. O gráfico mostrado na Figura 5.13 apresenta a lista das características mais importantes para o teste implementado.

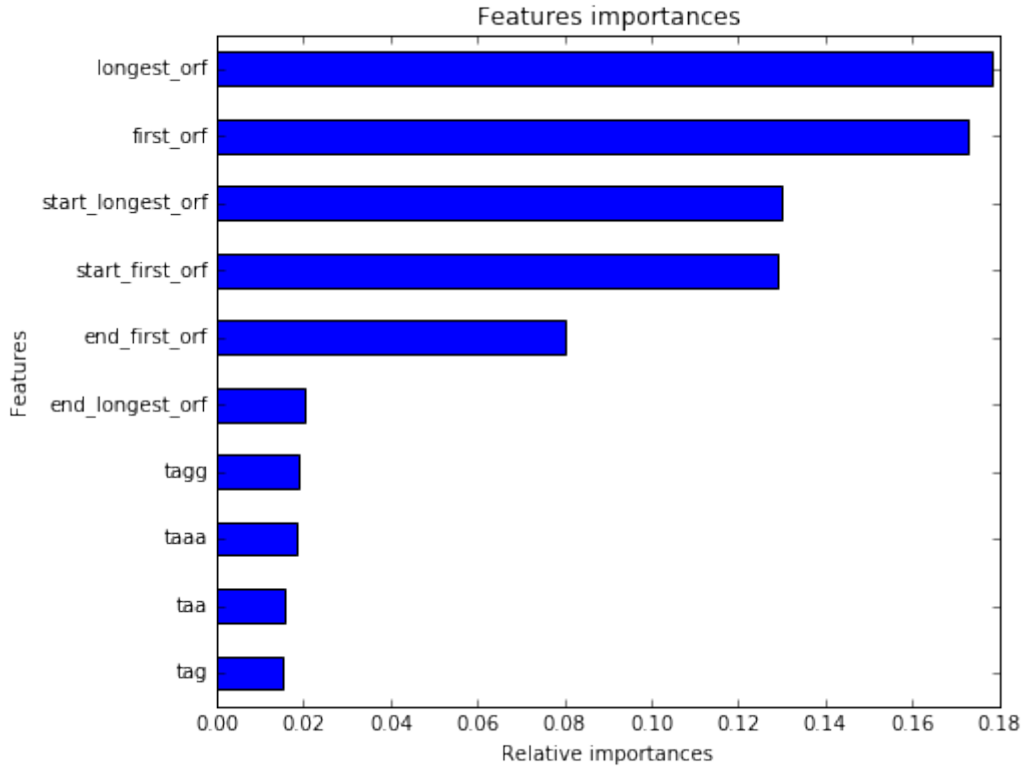


Figura 5.13: Teste 6 (PCTs Aleatórias): Importância relativa das características.

O gráfico da Figura 5.13 apresenta os tamanhos relativos da maior e primeira ORF seguido de suas posições de início e fim como as características mais importantes. Isso indica que essas características contribuem fortemente para o bom desempenho do modelo classificador.

Dados balanceados: PCTs selecionadas por método de clusterização

Nesta Seção, são apresentados os resultados obtidos utilizando dados balanceados, com PCTs selecionadas por método de clusterização [24]. O gráfico mostrado na Figura 5.14 apresenta a lista das características mais importantes para o teste implementado.

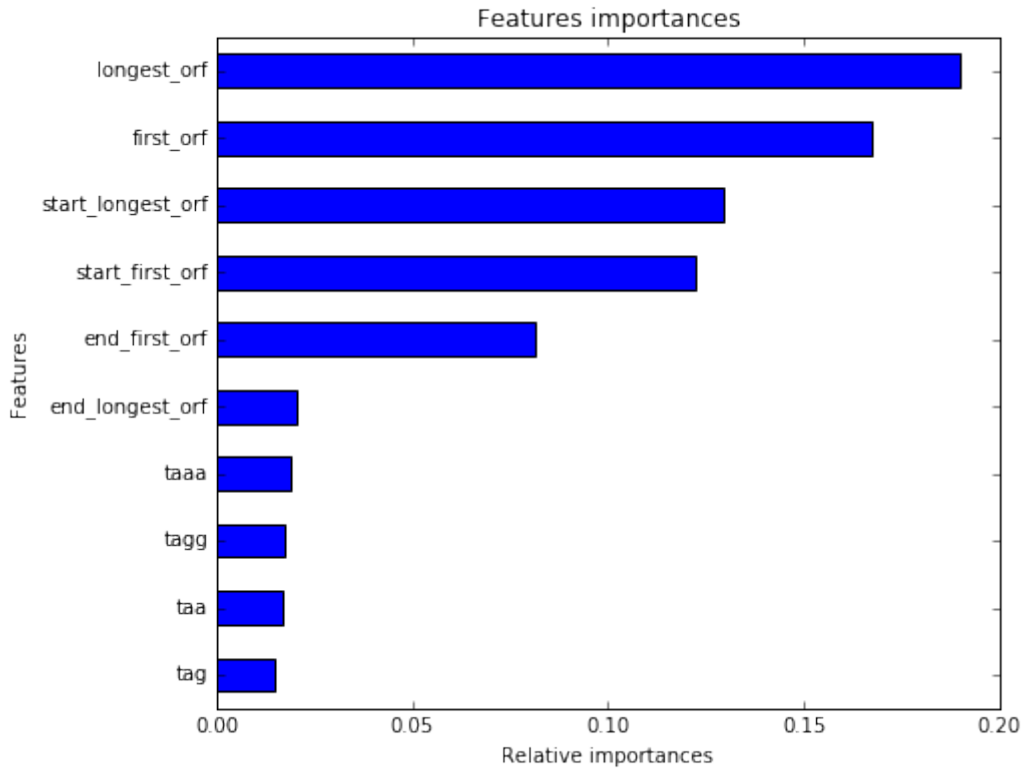


Figura 5.14: Teste 6 (PCTs Clusterizadas): Importância relativa das características.

O gráfico da Figura 5.14 apresenta os tamanhos relativos da maior e primeira ORF seguido de suas posições de início e fim como as características mais importantes. Isso indica que essas características contribuem fortemente para o bom desempenho do modelo classificador.

Dados desbalanceados: Dados apresentam mais PCTs

Nesta Seção, são apresentados os resultados obtidos com dados desbalanceados, apresentando mais PCTs. O gráfico mostrado na Figura 5.15 apresenta a lista das características mais importantes para o teste implementado.

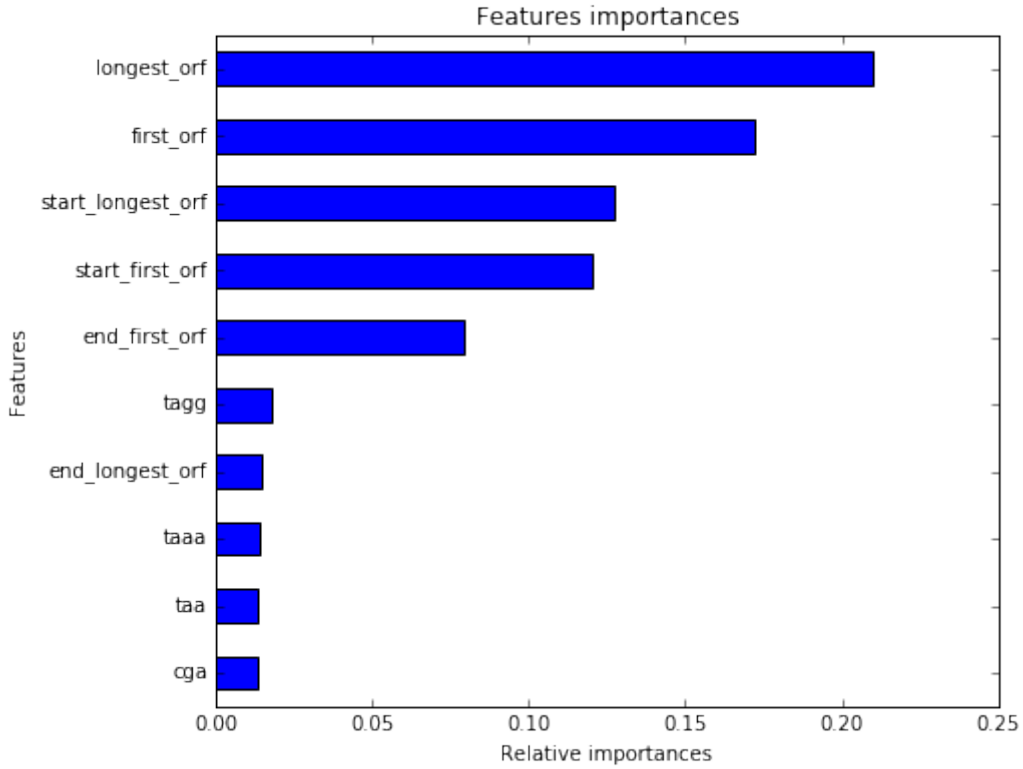


Figura 5.15: Teste 6 (Desbalanceado): Importância relativa das características.

O gráfico da Figura 5.15 apresenta os tamanhos relativos da maior e primeira ORF seguido de suas posições de início como as características mais importantes, apenas o tetra-nucleotídeo 'tagg' se mostrou mais importante do que a posição do fim da primeira ORF. Isso indica que essas características contribuem para o bom desempenho do modelo classificador.

5.4 Observações gerais

Nesta Seção observações gerais sobre os testes da Seção 5.2 serão apresentadas e analisadas. Nas seções 5.4.1 e 5.4.2 os resultados obtidos para os dados balanceados, em que as PCTs foram selecionadas aleatoriamente e por método de clusterização respectivamente, são analisados. Na Seção 5.4.3, os resultados obtidos para os dados desbalanceados são analisados. Na Seção 5.4.4, a *performance* do modelo preditivo utilizando o *Random Forest* é avaliado. Por último, na Seção 5.4.5 uma comparação com o método de Análise de Componentes Principais é realizada.

5.4.1 PCTs selecionadas aleatoriamente

Para melhor observar os resultados obtidos para os testes, em que os dados utilizados foram balanceados com PCTs selecionadas aleatoriamente, a Tabela 5.40 foi criada para apresentar a *performance* do modelo *Random Forest*, enquanto a Tabela 5.41 apresenta a *performance* do modelo SVM.

Os gráficos das Figuras 5.16 e 5.18 representam, respectivamente, as curvas das medidas estatísticas listadas na Seção 5.1 para os modelos *Random Forest* e SVM para conjuntos com apenas um grupo de características, enquanto os gráficos das Figuras 5.17 e 5.19 para conjuntos de dois ou mais grupos.

Tabela 5.40: *Performance* do modelo *Random Forest*.

Performance do modelo *Random Forest*					
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade	OOB
Teste 1	97	95	98	95	97
Teste 2	96	95	98	95	96
Teste 3	95	92	98	91	95
Teste 4	85	83	88	82	85
Teste 5	97	96	98	96	97
Teste 6	96	94	99	94	96

Um fato interessante apresentado na Tabela 5.40 é que a pontuação *out-of-bag*, representado pela coluna OOB, apresentou valores próximos à acurácia do modelo preditor. Isso mostra uma vantagem do modelo *Random Forest* em prover medidas estatísticas de boa qualidade já na construção do modelo. O fato da pontuação *out-of-bag* se aproximar do valor da acurácia do modelo confirma a qualidade da extração de características importantes dos lncRNAs provida pelo *Random Forest*, uma vez que os dados *out-of-bag* são diretamente utilizados na determinação das importâncias das características.

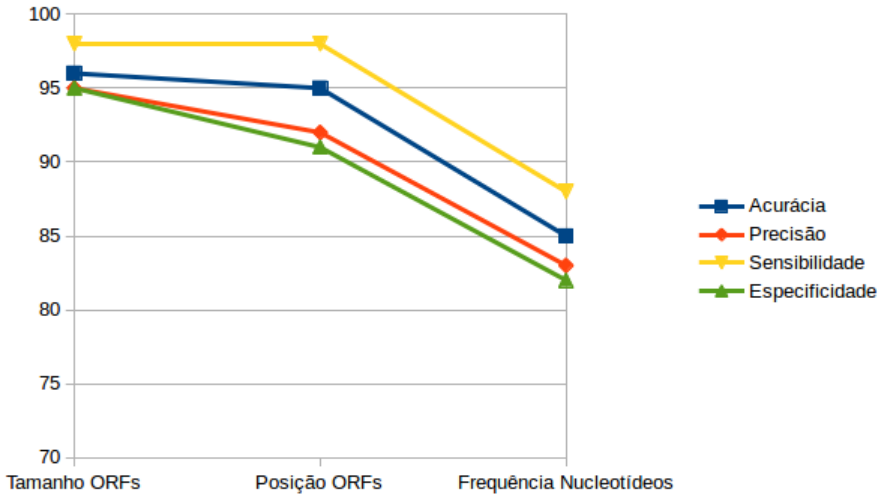


Figura 5.16: *Performance* do *Random Forest* para grupos com 1 característica.

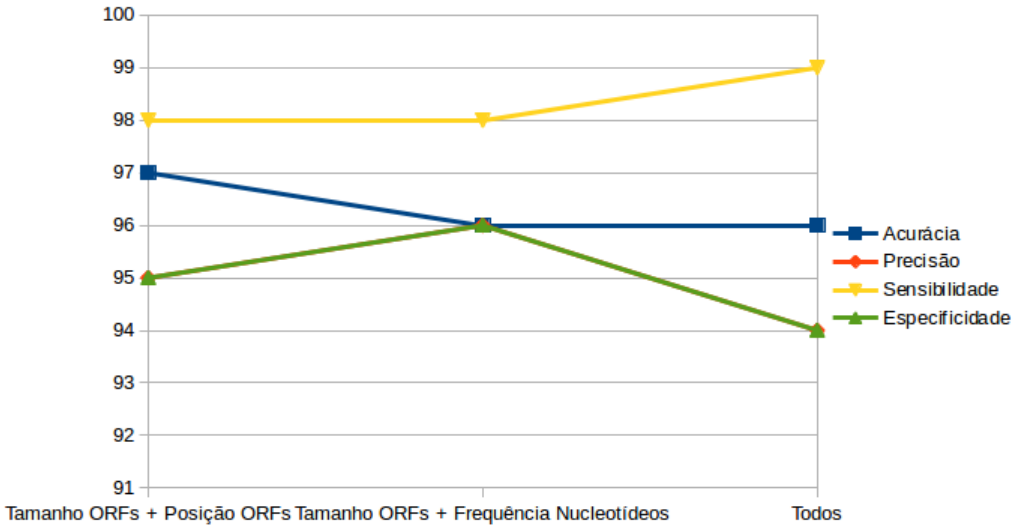


Figura 5.17: *Performance* do *Random Forest* para grupos com 2 ou mais características.

Os gráficos das Figuras 5.16 e 5.17 mostram que o *Random Forest* obteve um ótimo desempenho para classificar lncRNAs e PCTs corretamente. O modelo apresentou mais dificuldades no teste 4, em que apenas as características de frequências relativas dos di, tri e tetra-nucleotídeos foram utilizadas. Para os demais testes, as melhores *performances* obtidas foram para os que utilizaram os tamanhos relativos das ORFs entre as características.

Tabela 5.41: *Performance* do modelo SVM.

Performance do modelo SVM				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 1	95	94	96	94
Teste 2	96	96	97	96
Teste 3	92	87	99	85
Teste 4	88	88	88	88
Teste 5	97	96	98	96
Teste 6	93	97	90	97

Apesar de não apresentar resultados tão bons quanto os obtidos com o *Random Forest* em cinco dos seis testes, o modelo SVM monstrou uma boa *performance* em classificar lncRNAs e PCTs quando apenas as características de frequências relativas dos di, tri e tetra-nucleotídeos são utilizadas.

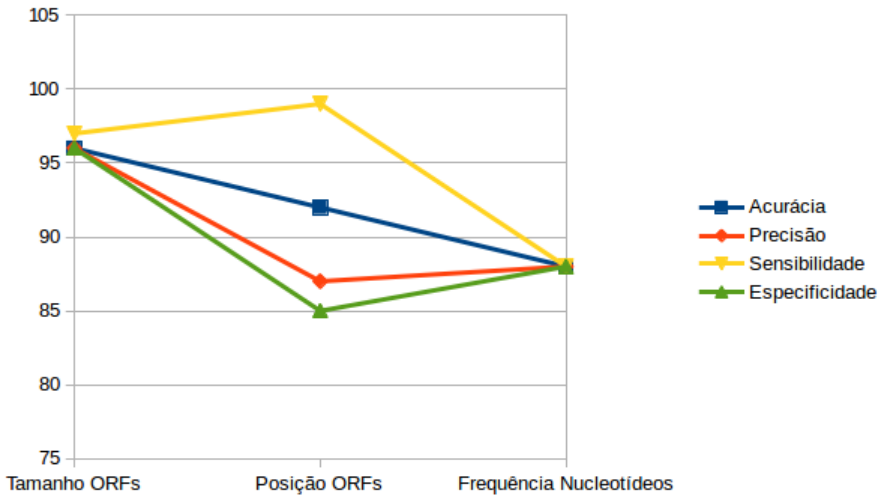


Figura 5.18: *Performance* do SVM para grupos com 1 característica.

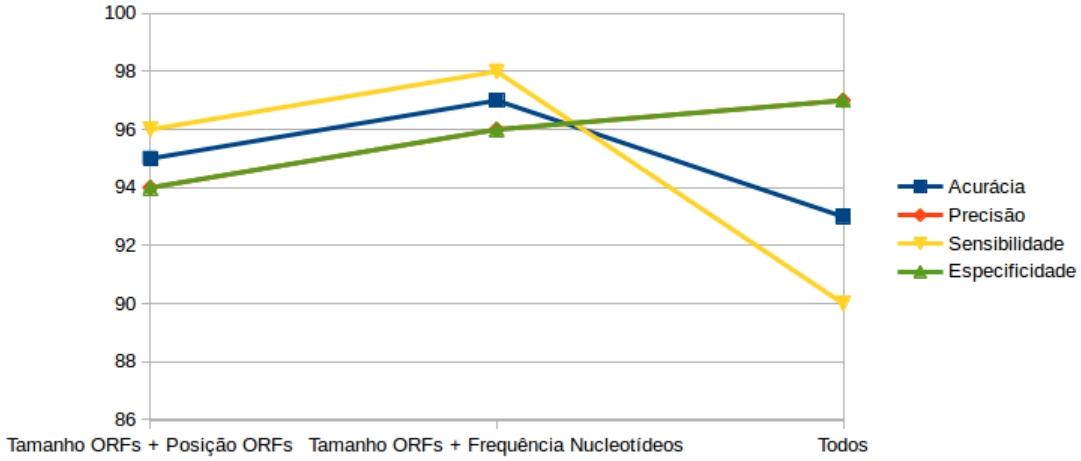


Figura 5.19: *Performance* do SVM para grupos com 2 ou mais características.

Os gráficos das Figuras 5.18 e 5.19 mostram que o *Random Forest* obteve um bom desempenho para classificar lncRNAs e PCTs corretamente. O modelo obteve sua pior *performance* no teste 4 em que apenas as características de frequências relativas dos di, tri e tetra-nucleotídeos foram utilizadas, entretanto os resultados se mostraram superiores aos obtidos pelo *Random Forest*. Para os demais testes, as melhores *performances* obtidas foram para os que utilizaram os tamanhos relativos das ORFs entre as características.

5.4.2 PCTs selecionadas por método de clusterização

Para melhor observar os resultados obtidos para os testes em que os dados utilizados foram balanceados com PCTs selecionadas por método de clusterização, a Tabela 5.42 foi criada para apresentar a *performance* do modelo *Random Forest*, enquanto a Tabela 5.43 apresenta a *performance* do modelo SVM.

Os gráficos das Figuras 5.20 e 5.22 representam, respectivamente, as curvas das medidas estatísticas listadas em 5.1 para os modelos *Random Forest* e SVM para conjuntos

com apenas um grupo de características, enquanto os gráficos das Figuras 5.21 e 5.23 para conjuntos de dois ou mais grupos.

Tabela 5.42: *Performance* do modelo *Random Forest*.

Performance do modelo *Random Forest*					
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade	OOB
Teste 1	99	99	99	99	99
Teste 2	97	95	98	95	97
Teste 3	94	91	98	91	94
Teste 4	85	83	88	82	85
Teste 5	97	97	98	96	97
Teste 6	99	99	99	99	99

A *performance* obtida pelo modelo utilizando dados balanceados com PCTs selecionadas por método de clusterização mostrou-se superior àquela obtida para modelos com dados balanceados com PCTs selecionadas aleatoriamente. Isso mostra o impacto dos dados utilizados no treinamento na *performance* geral do modelo.

A pontuação *out-of-bag* continuou a se aproximar da acurácia do sistema, obtendo bons resultados na extração de características importantes dos lncRNAs.

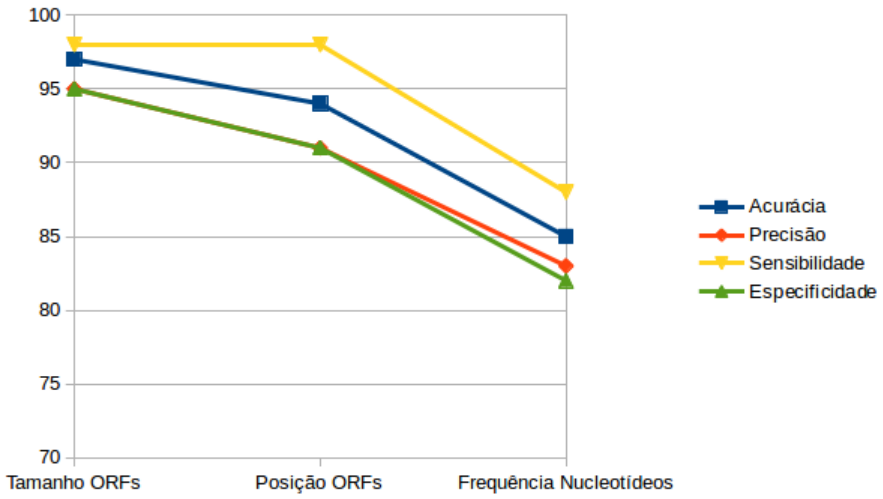


Figura 5.20: *Performance* do *Random Forest* para grupos com 1 característica.

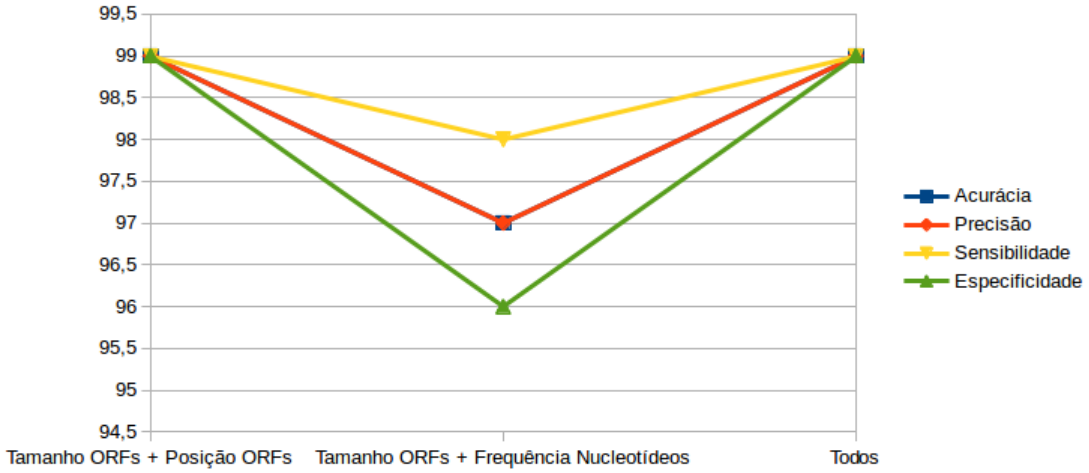


Figura 5.21: *Performance* do *Random Forest* para grupos com 2 ou mais características.

O modelo continuou apresentando uma *performance* melhor para os testes em que utilizaram as características de tamanho relativo das ORFs. O teste 4 continuou apresentando uma performance não tão boa.

Tabela 5.43: *Performance* do modelo SVM.

Performance do modelo SVM				
Conjunto	Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
Teste 1	97	99	95	99
Teste 2	97	96	98	96
Teste 3	88	82	99	78
Teste 4	88	89	88	89
Teste 5	97	97	98	97
Teste 6	93	88	99	87

Apesar de não apresentar resultados tão bons quanto os obtidos com o *Random Forest* em cinco dos seis testes, o modelo SVM monstrou uma boa *performance* em classificar lncRNAs e PCTs, quando apenas as características de frequências relativas dos di, tri e tetra-nucleotídeos são utilizadas. No entanto, como apontado pela especificidade, o modelo apresentou dificuldades ao classificar PCTs corretamente no teste 3, em que apenas características das posições de início e fim das ORFs são utilizadas.

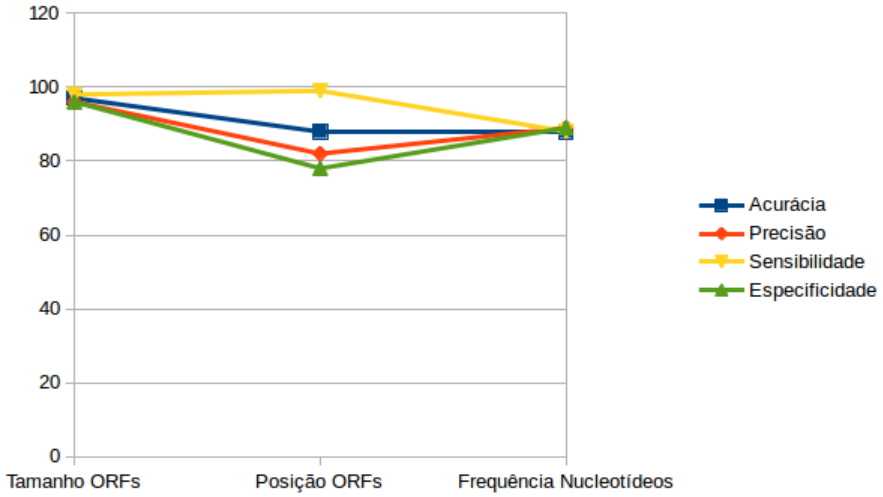


Figura 5.22: *Performance* do SVM para grupos com 1 característica.

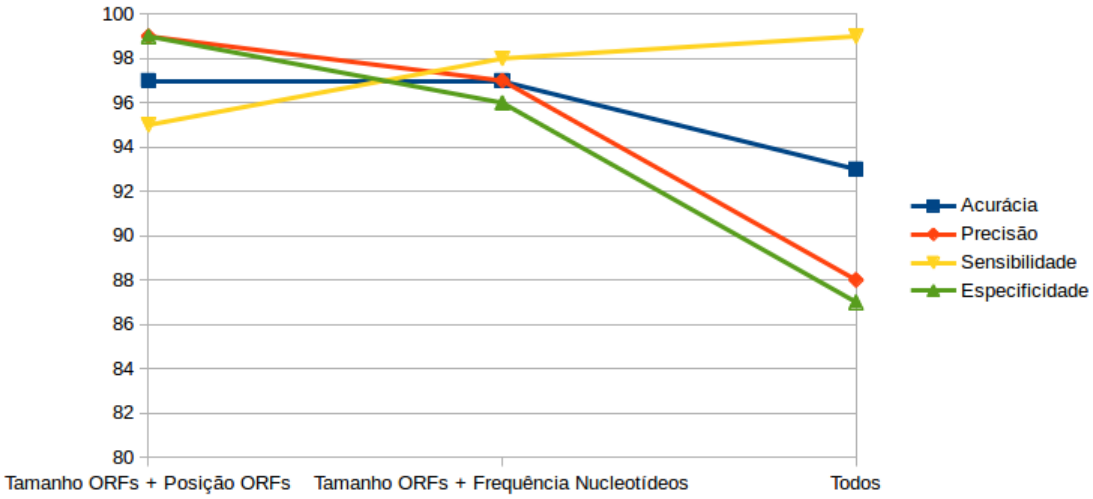


Figura 5.23: *Performance* do SVM para grupos com 2 ou mais características.

Os gráficos das Figuras 5.22 e 5.23 mostram que o *Random Forest* obteve um bom desempenho para classificar lncRNAs e PCTs corretamente. O modelo obteve sua pior *performance* no teste 4, entretanto os resultados mostraram-se superiores aos obtidos pelo *Random Forest*. Para os demais testes, as melhores *performances* obtidas foram para os que utilizaram os tamanhos relativos das ORFs entre as características.

5.4.3 Dados desbalanceados

Para melhor observar os resultados obtidos para os testes em que os dados utilizados foram balanceados com PCTs selecionadas aleatoriamente, a Tabela 5.44 foi criada para apresentar a *performance* do modelo *Random Forest*, enquanto a Tabela 5.45 apresenta a *performance* do modelo SVM.

Tabela 5.44: *Performance* do modelo *Random Forest*.

Performance do modelo *Random Forest*	
Conjunto	F-measure
Teste 1	92
Teste 2	92
Teste 3	88
Teste 4	53
Teste 5	93
Teste 6	93

A *performance* obtida pelo modelo utilizando dados desbalanceados se mostrou pior que aquelas que utilizaram dados balanceados. Isso mostra o impacto dos dados utilizados no trainamento na *performance* geral do modelo. O modelo apresentou um péssimo desempenho de classificação para o teste 4, em que apenas as características de frequências relativas dos di, tri e tetra-nucleotídeos foram utilizadas, mas obteve um bom desempenho para os demais testes.

Tabela 5.45: *Performance* do modelo SVM.

Performance do modelo SVM	
Conjunto	F-measure
Teste 1	81
Teste 2	92
Teste 3	72
Teste 4	76
Teste 5	94
Teste 6	73

Apesar de não apresentar bons resultados, o modelo SVM monstrou uma melhor *performance* que o *Random Forest* para classificar lncRNAs e PCTs utilizando apenas as características de frequências relativas dos di, tri e tetra-nucleotídeos.

5.4.4 *Performance* do *Random Forest*

Para poder testar a *performance* do modelo *Random Forest*, foram feitas comparações dos resultados obtidos pelo SVM. Para isso foi criado um gráfico das acurácias de cada um dos modelos para os dados de entrada balanceados (aleatórios e clusterizados) e desbalanceados como mostram as Figuras 5.24, 5.25 e 5.26.

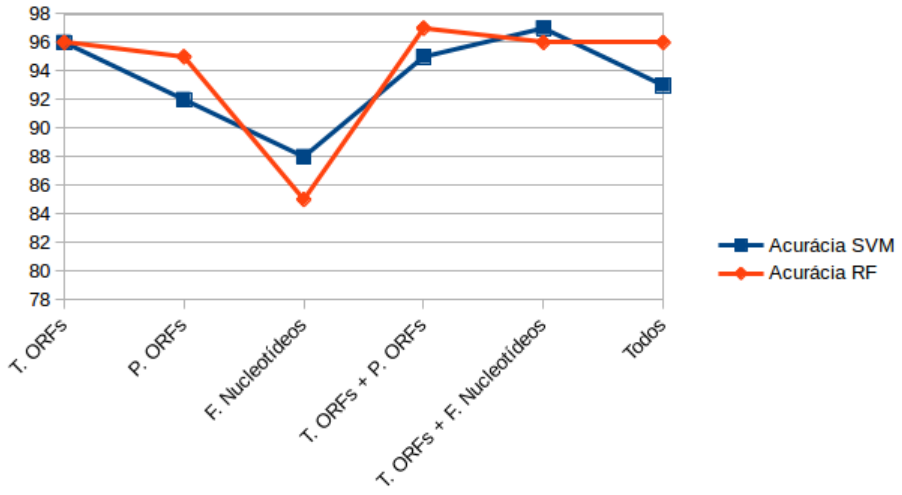


Figura 5.24: Comparação da acurácia de dados balanceados com PCTs selecionadas aleatoriamente nos modelos *Random Forest* e SVM.

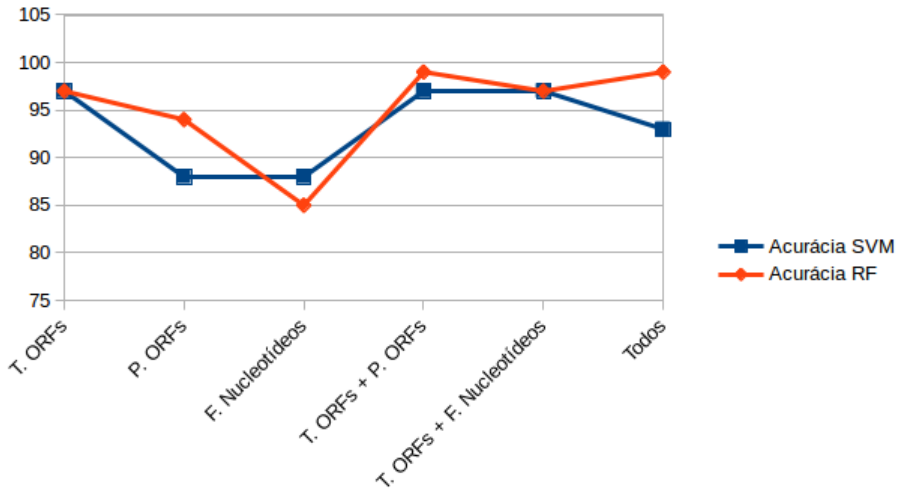


Figura 5.25: Comparação da acurácia de dados balanceados com PCTs clusterizadas nos modelos *Random Forest* e SVM.

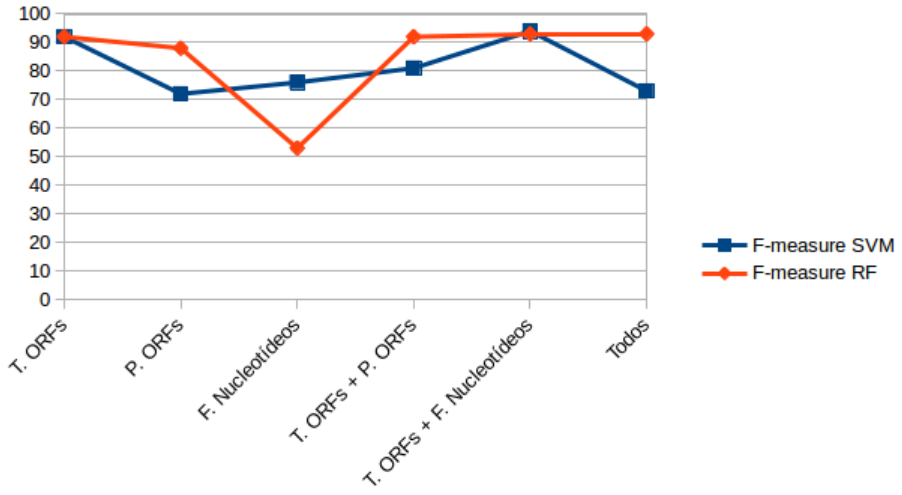


Figura 5.26: Comparação da acurácia de dados desbalanceados nos modelos *Random Forest* e SVM.

É possível observar que, para todos os tipos de dados de entrada cinco dos seis casos de testes da Seção 4.2 o *Random Forest* obteve uma *performance* muito próxima ou melhor que o SVM.

O único caso que se mostrou superior no modelo SVM foi o caso de teste em que as frequências relativas dos di,tri e tetra-nucleotídeos são as únicas características levadas em consideração na construção do modelo. Isso pode ocorrer pelo fato de uma seleção aleatória de características poder não funcionar propriamente, já que características não-informativas ou correlacionadas podem ser selecionadas constantemente para a montagem das árvores na floresta, o que pode degradar o desempenho do classificador.

Apesar da presença dos tamanhos relativos das ORFs elevar o desempenho preditivo do modelo, o teste 5 do modelo apresentou uma *performance* um pouco inferior à do SVM nos casos de dados balanceados com PCTs selecioandas aleatoriamente e dados desbalanceados. Isso deve-se ao fato da má *performance* do *Random Forest* ao utilizar as características das frequências relativas dos di,tri e tetra-nucleotídeos, o que diminui um pouco qualidade preditiva do modelo.

O *Random Forest* apresentou uma boa *performance* para o teste 3 para dados balanceados e desbalanceados. Como não foi encontrado na literatura indícios de que as posições de início e fim das ORFs são relevantes para a classificação dos lncRNAs, é levantada a hipótese de que o modelo construído com esse teste pode estar apresentando *overfitting*. Isso ocorre quando a classificação funciona bem apenas para um determinado conjuntode dados. As PCTs também pode estar apresentando dados correlacionados. As PCTs, que foram utilizados como conjunto negativo do modelo, podem apresentar dados semelhantes, como posições de início e fim das ORFs, o que pode elevar a importância dessas características e consequentemente a qualidade de predição do modelo.

Um fato importante sobre as PCTs é que grande parte dos bacos de dados [26, 28] apresentam o início de seus transcritos a partir do início da ORF, ou seja, grande parte dos dados das PCTs podem estar apresentando o início da ORF na posição 1 do transcrito e apresentando sua posição final da ORF no fim do transcrito. Além disso, isso pode fazer com que a maior ORF seja igual a primeira ORF, uma vez que o transcrito em si é a

primeira ORF e possui seu tamanho igual ao tamanho do transcrito, causando assim a equivalência entre primeira e maior ORF.

Para os dados balanceados e desbalanceados, o *Random Forest* apresentou sua melhor *performance* quando todas as característica da Seção 4.1.1 foram utilizadas em conjunto. Essa melhora no modelo preditivo deve-se ao fato de que a inclusão de características mais importantes melhora a generalização do modelo criado. Além disso, é possível observar que as *performances* do *Random Forest* para teste 6 foram muito superiores às obtidas pelo SVM.

5.4.5 Comparação das caracteristicas encontradas no modelo *Random Forest* com o método PCA

Para poder testar a *performance* do modelo *Random Forest* ao determinar a importância dos di e tri e tetra-nucleotídeos dos lncRNAs, foram comparadas as 50 e 60 combinações de nuceotídeos mais importantes encontradas para os tipos de dados balanceados (aleatórios e clusterizados) e desbalanceados no modelo *Random Forest* com os dados encontratos pelo Schneider [70] que utilizou um método estatístico chamando Análise de Componentes Principais (*Principal component analysis - PCA*) que é um método que realiza a análise dos dados usados visando sua redução, eliminação de sobreposições e a escolha das formas mais representativas de dados a partir das combinações lineares das variáveis originais.

A Tabela 5.46 apresenta os 50 di e tri e tetra-nucleotídeos encontrados como sendo os mais importantes pelo método PCA.

Tabela 5.46: 50 frequências mais importantes pelo método PCA.

Ranking	Sequências
1-10	aa, tt, cc, gg, ccc, ggg, ttt, aaa, aca, ata
11-20	gtg, tct, atg, tat, cag, cac, aga, ctc, tca, tgt
21-30	gag, at, ctg, cat, ag, tga, ta, ca, tg, ct
31-40	ac, cta, cgc, tc, gt, ga, gcg, cg, gc, act
41-50	att, tag, gtc, caa, tac, atc, ttg, gac, acg, gta

A Tabela 5.47 apresenta os 60 di e tri e tetra-nucleotídeos encontrados como sendo os mais importantes pelo método PCA.

Tabela 5.47: 60 frequências mais importantes pelo método PCA.

Ranking	Sequências
1-10	aa, tt, cc, gg, ctgg, ccc, ggg, ttt, aaa, aaaa
11-20	caga, tgga, aaga, gaga, cagc, cctg, aca, ata, gtg, cagg
21-30	gaag, tct, atg, cac, ctc, tat, cag, aga, gag, ctg
31-40	tgt, tca, at, cat, cta, ag, cgc, tga, tg, ca
41-50	ta, ct, gcg, tc, ac, tag, ga, gt, gc, cg
51-60	act, cca, tac ,tcg, att, gtc, tgg, caa, gac, ttg

Comparando os dados das Tabelas 5.19, 5.21 com a Tabela 5.46 foi constatado que todas elas possuem em comum 11 di, tri e tetra-nucleotídeos. São esses: 'tt', 'ttt', 'aaa', 'tct', 'atg', 'ta', 'ct', 'ga', 'cg', 'att', 'tag'. Quando comparado com a Tabela 5.39 foi constatado a presença de 11 nucleotídeos em comum, porém foi observado a presença dos transcritos 'cac' e 'ctc' e ausência dos transcritos 'ta' e 'att'.

Já para a Tabela 5.47 comparada com as Tabelas 5.19, 5.21 e 5.39 foi constatado que todas elas possuem em comum 17 di, tri e tetra-nucleotídeos. São esses: 'tt', 'ttt', 'aaa', 'tgga', 'gaag', 'tct', 'atg', 'ctc', 'ta', 'ct', 'tag', 'ga', 'cg', 'tac', 'tcg', 'att' e 'tgg'.

Com essa comparação podemos dizer que essas sequências de nucleotídeos apresentam fortes indícios de que podem ser relevantes para a classificação dos lncRNAs. O método PCA utilizado pelo Schneider [70] também apontou o tamanho relativo da primeira ORF do transcrito como uma característica importante para a classificação dos lncRNAs.

5.5 Criação de modelo preditivo utilizando as características mais importantes

Nesta Seção será proposto a criação de um modelo preditivo utilizando as características mais importântes obtidas na Seção 5.2.

As características mais importântes de cada teste da Seção 4.2 foram selecionadas para a criação de modelos preditivos. Um modelo preditivo com características mais informativas tende a apresentar uma melhor *performance*.

Os dados utilizados para a contrução desse modelo foram aqueles em que as PCTs foram selecionadas por método de clusterização [24].

Neste este trabalho, são propostos quatro testes para o modelo preditivo. Para os dois primeiros, os 11 e 17 di, tri e tetra-nucleotídeos obtidos na Seção 5.4.5 foram utilizados como as únicas características do modelo. A Seção 5.5.1 apresenta as *performances* destes modelos. Para os últimos dois testes, o tamanho relativo e as posições de início e fim da primeira e maior ORF além dos respectivos 11 e 17 di, tri e tetra-nucleotídeos mais importantes, foram as caraterísticas utilizadas nos modelos. A Seção 5.5.2 apresenta a *performances* destes modelos.

Para treinamento e teste do modelo, as PCTs foram selecionadas por método de clusterização [24]. Para treinamento do modelo, 20.000 PCTs e 20.000 lncRNAs foram utilizados. Para teste, 5.000 PCTs e 5.000 lncRNAs foram utilizados.

5.5.1 Modelo preditivo utilizando os di, tri e tetra-nucleotídeos mais importantes

As Tabelas 5.48 e 5.49 apresentam as *performances* do modelo quando apenas as frequências dos 11 nucleotídeos mais importantes foram utilizadas.

Tabela 5.48: Teste com os 11 di, tri e tetra-nucleotídeos mais importantes.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	3838	1162
lncRNA	962	4038
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	3653	1347
lncRNA	1365	3635

Tabela 5.49: *Performance* dos modelos *Random Forest* e SVM com as 11 frequências mais importantes.

Performance do modelo *Random Forest*			
Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
79	78	81	77
Performance do modelo *SVM*			
Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
73	73	73	73

É possível perceber pelas Tabelas 5.48 e 5.49 que o *Random Forest* apresentou uma *performance* superior a do SVM utilizando as características dos 11 di, tri e tetra-nucleotídeos como as únicas do modelo.

As Tabelas 5.50 e 5.51 apresentam as *performances* do modelo quando apenas as frequências dos 17 nucleotídeos mais importantes foram utilizadas.

Tabela 5.50: Teste com os 17 di, tri e tetra-nucleotídeos mais importantes.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	3956	1044
lncRNA	842	4158
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	3962	1038
lncRNA	1118	3882

Tabela 5.51: *Performance* dos modelos *Random Forest* e SVM com as 17 frequências mais importantes.

Performance do modelo *Random Forest*			
Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
81	80	83	79
Performance do modelo *SVM*			
Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
78	79	78	79

É possível perceber pelas Tabelas 5.50 e 5.51 que o *Random Forest* apresentou uma *performance* superior a do SVM utilizando as características dos 17 di, tri e tetra-nucleotídeos como as únicas do modelo. O *Random Forest* apresentou uma maior facilidade ao predizer dados de lncRNAs corretamente, como é mostrado por sua sensibilidade.

Com a comparação das Tabelas 5.49 e 5.51 é possível observar que o modelo construído com as características dos 17 di, tri e tetra-nucleotídeos apresentou melhor desempenho do que o que utilizou apenas os 11 nucleotídeos mais importantes. Isso indica que os 6 nucleotídeos ausentes no modelo apresentam uma alta importância relativa, capaz de elevar o desempenho de um modelo preditivo.

Nota-se que apesar de superior ao SVM, para ambos os testes, o modelo não obteve uma *performance* superior a obtida no teste 4 em que todas as frequências relativas dos di, tri e tetra-nucleotídeos foram utilizadas. Isso deve-se ao fato do teste 4 apresentar não apenas os 11 ou 17 di, tri e tetra-nucleotídeos encontrados em comum com o método PCA da Seção 5.4.5, mas também os outros nucleotídeos mais importantes, como mostra a Tabela 5.38. Sendo assim, a ausência de características mais informativas reduziu a *performance* do modelo.

5.5.2 Modelo preditivo utilizando todas as características mais importantes

As Tabelas 5.52 e 5.53 apresentam as *performances* do modelo quando as características do tamanho da primeira e maior ORF, suas posições de início e fim e os 11 di, tri e tetra-nucleotídeos mais importantes foram utilizadas.

Tabela 5.52: Teste com todas as características mais importantes.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4961	39
lncRNA	73	4927
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4906	94
lncRNA	133	4867

Tabela 5.53: *Performance* dos modelos *Random Forest* e SVM com todas as características mais importantes.

Performance do modelo *Random Forest*			
Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
99	99	99	99
Performance do modelo *SVM*			
Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
98	98	97	98

É possível perceber pelas Tabelas 5.52 e 5.53 que o *Random Forest* apresentou uma *performance* pouco superior a do SVM utilizando todas as características mais importantes no modelo.

As Tabelas 5.54 e 5.55 apresentam as *performances* do modelo quando as características do tamanho da primeira e maior ORF, suas posições de início e fim e os 17 di, tri e tetra-nucleotídeos mais importantes foram utilizadas.

Tabela 5.54: Teste com todas as características mais importantes.

Predição do modelo *Random Forest*		
Valor real	PCT	lncRNA
PCT	4961	39
lncRNA	84	4916
Predição do modelo SVM		
Valor real	PCT	lncRNA
PCT	4749	251
lncRNA	94	4906

Tabela 5.55: *Performance* dos modelos *Random Forest* e SVM com todas as características mais importantes.

Performance do modelo *Random Forest*			
Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
99	99	98	99
Performance do modelo *SVM*			
Acurácia (%)	Precisão (%)	Sensibilidade (%)	Especificidade (%)
97	95	98	95

É possível perceber pelas Tabelas 5.54 e 5.55 que o *Random Forest* apresentou uma *performance* superior a do SVM utilizando todas as características mais importantes no modelo. O *Random Forest* apresentou uma maior acurácia e precisão além de uma maior facilidade ao predizer PCTs corretamente, como é apontado por sua especificidade.

Observa-se, comparando as tabelas e 5.53 5.55, que se diferenciam pela presença dos 11 e 17 di, tri e tetra-nucleotídeos, que as performances dos modelos utilizando foi muito semelhante. Isso indica que a presença dos tamanhos relativos da maior e primeira ORF além de suas posições de início e fim são características que apresentam um alto poder informativo o que eleva o desempenho do modelo preditivo.

Nota-se que o modelo também apresentou uma *performance* melhor que a do teste 6, em que todas as características foram utilizadas. Isso deve-se ao fato de características menos informativas tendem a reduzir a qualidade preditiva do modelo.

Capítulo 6

Conclusão

Neste trabalho, foi proposto um método para extração de características importante para predição de lncRNAs baseado no algoritmo *Random Forest*. Foi desenvolvido um estudo de caso com foco na extração de características e na construção de um modelo preditivo de classificação para os lncRNAs em humanos.

O tamanho relativo das ORFs, posições de início e fim das ORFs, além das frequências relativas dos di, tri e tetra-nucleotídeos foram as características utilizadas na construção do modelo. Vários testes foram aplicados utilizando essas características o que permitiu verificar a importância do tamanho relativo das ORFs na classificação dos lncRNAs. Além dessa característica, nosso método também apontou as posições de início e fim da maior ORF e da primeira ORF como características importantes para a classificação dos lncRNAs.

Essas características - tamanho relativo das ORFs, posições de início e fim da maior ORF e da primeira ORF - apresentaram boa acurácia quando incluídas no conjunto de características do modelo. Essa descoberta não havia sido encontrada na literatura até então, o que levanta a possibilidade do modelo apresentar *overfitting*. Porém, o mesmo método poderia ser aplicado para um determinado organismo de interesse, e essas características encontradas pelo modelo *Random Forest* poderiam ser utilizadas no modelo de predição de lncRNAs.

Existe também a possibilidade das posições de início e fim das ORFs terem sido apontadas como importantes devido a presença de dados correlacionados. As PCTs, que foram utilizados como conjunto negativo do modelo, apresentam dados semelhantes, como posições de início e fim das ORFs, o que pode elevar a importância dessas características e consequentemente a qualidade de predição do modelo.

Foi possível observar que o modelo preditivo do *Random Forest* apresentou uma performance melhor que o SVM. Para dados balanceados, com todas as características, apresentou uma acurácia de 96% quando as PCTs foram selecionados aleatoriamente e 99% quando as PCTs selecionadas foram clusterizadas, enquanto o SVM apresentou 93% para ambos os casos. Para dados desbalanceados (com maior número de PCTs), também apresentou uma boa performance com um *F-measure* de 93%, enquanto o SVM apresentou 73%.

Comparando os resultados das 60 combinações de nucleotídeos mais relevantes apontadas pelo *Random Forest* com o método estatístico PCA [70], 17 di, tri e tetra-nucleotídeos listados na seção 5.4.5 parecem ser relevantes para a classificação de lncRNA.

Na construção do modelo preditivo utilizando apenas os 17 di, tri e tetra-nucleotídeos mais importantes como suas únicas características, o *Random Forest* apresentou uma acurácia de 81%, superior aos 78% apresentado pelo SVM.

No entanto, a melhor acurácia obtida para o modelo foi quanto as características mais importantes obtidas foram utilizadas. Com os tamanhos relativos das ORFs, posições de início e fim das ORFs e os 17 di, tri e tetra-nucleotídeos mais importantes, o *Random Forest* apresentou uma acurácia de 99%, superior aos 97% apresentado pelo SVM.

Neste trabalho, o *Random Forest* mostrou que, além de ser um bom algoritmo para a identificação de características que parecem importantes para a classificação dos lncRNAs, também pode ser utilizado para construir modelos preditivos de lncRNAs com uma boa acurácia.

6.1 Contribuições

Neste projeto, fizemos duas contribuições relevantes:

- Propusemos um modelo para extração de características para a predição de lncRNAs baseado no método de aprendizagem de máquina *Random Forest*;
- Além das características já conhecidas na literatura, identificamos as posições de início e fim da maior ORF e da primeira ORF, além de 17 nucleotídeos, como possíveis características importantes para a classificação de lncRNAs;
- Criação de um modelo preditivo de boa acurácia para lncRNAs baseado no método de aprendizagem de máquina *Random Forest*;

6.2 Trabalhos futuros

Os próximos trabalhos a serem realizados:

- Escrever um artigo com o modelo do *Random Forest* criado neste projeto;
- Realizar testes de validação, com outros mamíferos, para testar a capacidade de generalização do modelo *Random Forest*;
- Refinar a seleção do conjunto de dados negativos para treino e teste do modelo, para garantir PCTs diversificadas, de forma a aumentar a generalização do modelo preditivo e garantir que não ocorra *overfitting* e presença de dados correlacionados.

Referências

[1] S. Ananiadou and J. Mcnaught. Text mining for biology and biomedicine. *Norwood, MA: Artech House*, 2006. 23

[2] W. Arbex, N. F. Martins, and M. F. Martins. *Talking About Computing and Genomic TACG - Modelos e Métodos Computacionais em Bioinformática*, volume 1. Embrapa, Brasília, DF, 2014. 32, 35, 36, 37

[3] W. C. Arruda. *ncRNA-Agents: Anotação de RNAs não-codificadores Baseada em Sistema Multiagente*. PhD thesis, Universidade de Brasília, Campus Universitário Darcy Ribeiro, Brasília - DF, 70910-900, 12 2015. Tese (Doutorado em Informática). 17, 19, 21

[4] H. V. Bakel, C. Nislow, B. J. Blencowe, and T. R. Hughes. Most "dark matter"transcripts are associated with known genes. *PLoS Biol*, 8(5):5, 2010. 15

[5] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002. http://www.accrue.com/products/rp_cluster_review.pdf. 25

[6] Pequenos Biólogos. Molécula de DNA. https://pequenosbiologos.files.wordpress.com/2010/09/dna.jpg, 2016. [Online; accessed 08-may-2016]. 8

[7] S. Boltaña, D. Valenzuela-Miranda, A. Aguilar, S. Mackenzie, and C. Gallardo-Escárate. Long noncoding RNAs (lncRNAs) dynamics evidence immunomodulation during ISAV-Infected Atlantic salmon (Salmo salar). *Scientific Reports*, 6, 4 2016. 65

[8] L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 8 1996. 31, 32

[9] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. 28

[10] G. Chen, Z. Wang, D. Wang, C. Qiu, M. Liu, X. Chen, O. Zhang, G. Yan, and O. Cui. LncRNADisease: a database for long-non-coding RNA-associated diseases. *Nucleic Acids Research*, 41(Database-Issue):983–986, 2013. 22

[11] F. Clésio. Data Mining - MATRIZ DE CONFUSÃO. https://mineracaodedados.wordpress.com/tag/matriz-de-confusao/, 2016. [Online; accessed 20-nov-2016]. 46

[12] Coladaweb. Transcrição do DNA em RNA. http://www.coladaweb.com/files/transcricao(1).jpg, 2016. [Online; accessed 08-may-2016]. 13

[13] G. V. Dantas. *Utilização de classificador Random Forest na detecção de falhas em Máquinas Rotativas*. Escola Politécnica, Universidade Federal do Rio de Janeiro, Av. Athos da Silveira Ramos, 149 - Bloco A - Cidade Universitária, Rio de Janeiro - RJ, 21941-909, 8 2015. Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação. 37

[14] Casa das Ciencias. Aminoácidos. http://wikiciencias.casadasciencias.org/wiki/images/thumb/7/79/Aminoacido_figura_1.png/250px-Aminoacido_figura_1.png, 2016. [Online; accessed 08-may-2016]. 10

[15] Casa das ciências. Tradução do RNAm em proteína. http://wikiciencias.casadasciencias.org/wiki/index.php/Tradu%C3%A7%C3%A3o, 2016. [Online; accessed 08-may-2016]. 13

[16] C. L. de Castro and A. P. Braga. Supervised learning with imbalanced data sets: an overview. *Sba Controle & Automação*, 22(5):441–466, 2011. 24

[17] L. H. S. de Lelis. Aprendizagem Semi-Supervisionada aplicada à Engenharia Financeira. Master's thesis, Universidade Federal de Minas Gerais, Av. Pres. Antônio Carlos, 6627 - Pampulha, Belo Horizonte - MG, 31270-901, 6 2007. Programa de Pós-graduação em Engenharia Elétrica da UFMG como requisito parcial para obtenção do grau de mestre em Engenharia Elétrica. 26, 27

[18] Diana. Diana. http://diana.imis.athena-innovation.gr/DianaTools/index.php, 2016. [Online; accessed 15-oct-2016]. 22

[19] J. Donaldson. Funcionamento de uma árvore de decisão. https://blog.bigml.com/2012/01/23/beautiful-decisions-inside-bigmls-decision-trees/, 2016. [Online; accessed 04-jun-2016]. 33

[20] Associação Nacional dos Inventores. Sequenciamento do DNA da Bactéria Chromobacterium Violaceum. http://inventores.com.br/sequenciamento-do-dna-da-bacteria-chromobacterium-violaceum/, 2010. [Online; accessed 08-may-2016]. 2

[21] P. Clote e R. Backofen. *Computational Molecular Biology: An Introduction.* John Willey sons Ltd, 2000. 2, 6, 12

[22] S. Eddy. Non-coding RNA genes and the modern RNA world. *Reviews Genetics*, 2(12):919—-929, 2001. 13, 14

[23] Edoceo. CSV. http://edoceo.com/utilitas/csv-file-format, 2016. [Online; accessed 08-may-2016]. 43

[24] EMBL-EBI. Clustalo Omega. https://www.ebi.ac.uk/Tools/msa/clustalo/, 2016. [Online; accessed 19-nov-2016]. 43, 49, 51, 54, 56, 59, 62, 65, 68, 71, 74, 76, 79, 92, 93

[25] EMBL-EBI. Rfam. http://rfam.xfam.org/, 2016. [Online; accessed 19-nov-2016]. 20

[26] EMBL-EBI and Wellcome Trust Sanger Institute. Ensembl. http://www.ensembl.org/index.html, 2016. [Online; accessed 15-oct-2016]. 21, 46, 90

[27] EMBL-EBI and Wellcome Trust Sanger Institute. Fasta. http://ensemblgenomes.org/info/access/ftp, 2016. [Online; accessed 15-oct-2016]. 41

[28] EMBL-EBI and Wellcome Trust Sanger Institute. Havana. http://vega.sanger.ac.uk/index.html, 2016. [Online; accessed 15-oct-2016]. 21, 46, 90

[29] Pesquisa FAPESP. Xylella – Concluído o genoma da bactéria. http://revistapesquisa.fapesp.br/2000/02/01/xylella-concluido-o-genoma-da-bacteria/, 2 2000. [Online; accessed 08-may-2016]. 2

[30] Maria Sueli Soares Felipe. Genoma Funcional e Diferencial do Paracoccidioides brasiliensis- Rede Genoma Centro-Oeste. https://page.ucb.br/bc/pesquisador.detalhes?idc=38378, 2016. [Online; accessed 08-may-2016]. 2

[31] The National Center for Biotechnology. BLAST. https://blast.ncbi.nlm.nih.gov/Blast.cgi, 2016. [Online; accessed 21-nov-2016]. 19

[32] Indian Assiciation for the Cultivation of Science. lnCeDB. http://gyanxet-beta.com/lncedb/browse_data.php?found=1&id=1, 2016. [Online; accessed 19-nov-2016]. 22

[33] Genomasur. ÁCIDOS NUCLEICOS. http://genomasur.com/lecturas/02-36-G.gif, 2016. [Online; accessed 08-may-2016]. 7

[34] G.J. Hannon, F.V. Rivas, and E.P. Murchison. The expanding universe of noncoding RNAs. *Cold Spring Harb Symp Quant Biol*, 71:551—-564, 2006. 16

[35] S. Haykin. *Neural Networks: A Comprehensive Foundation*, volume xii,xiv. Prentice Hall, 1999. 28

[36] I. L. Hofacker. Vienna RNA secondary structure server. Technical report, Institute for Theoretical Chemistry University of Vienna, Wahringerstrabe 17, 1090 Wien, Austria, 2003. http://rna.tbi.univie.ac.at/. 21

[37] Infoescola. Os 20 aminoacidos essenciais ao organismo. http://www.infoescola.com/bioquimica/os-20-aminoacidos-essenciais-ao-organismo/, 2016. [Online; accessed 08-may-2016]. 11

[38] InfoEscola. Replicação de DNA. http://www.infoescola.com/wp-content/uploads/2007/10/replicacao-de-DNA.jpg, 2016. [Online; accessed 08-may-2016]. 12

[39] National Human Genome Research Institute. Transfer RNA (tRNA). https://www.genome.gov/dmd/img.cfm?node=Photos/Graphics&id=85250, 2016. [Online; accessed 08-may-2016]. 18

[40] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. pages 137–142, 1998. 28

[41] V. S. José. *Projeto Genoma Humano: Utopia do homem geneticamente perfeito.*, volume 1. Edições Loyola, 2004. 1

[42] G. M. O. Junior. *Máquina de Vetores Suporte: estudo e análise de parâmetros para otimização de resultado.* Universidade Federal de Pernambuco, Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife - PE, 50670-901, 12 2010. Graduação em Ciências da Computação. 29, 30

[43] R. C. Junior. *Uso da mineração de dados na identificação de alunos com perfil de evasão do ensino superior.* Universidade de Santa Cruz do Sul, Rua da Garoupa, s/n - Capão Novo, Capão da Canoa - RS, 95555-000, 2015. Graduação em Ciências da Computação. 33, 34, 35

[44] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996. 27

[45] J. Koolman and K. H. Roehm. *Color Atlas of Biochemistry*, volume 2. Georg Thieme Verlag, 2005. 6

[46] H. Lodish, A. Berk, and P. Matsudaira. Molecular Cell Biology. *W. H. Freeman Co*, 2005. 9

[47] L. Lopes. Aprendizagem de máquina baseada na combinação de classificadores em bases de dados da área de saúde. Master's thesis, Pontifícia Universidade Católica do Paraná, R. Imac. Conceição, 1155 - Prado Velho, Curitiba - PR, 80215-901, 2007. 31

[48] R. Z. Lopez. *Classificação automática de defeitos em máquinas rotativas.* Universidade Federal do Rio de Janeiro, Av. Pedro Calmon, 550 - Cidade Universitária, Rio de Janeiro - RJ, 21941-901, 12 2014. Graduação em Ciências da Computação. 38

[49] R. Geslain Q. Dai M.R. Rosner M. Pavon-Eternod, S. Gomes and T. Pan. tRNA over-expression in breast cancer and functional consequences. *Nucleic Acids Research*, 37(21):7268–7280, 2009. 16

[50] A. Machado-Lima, H. Del Portillo, and A. Durham. Computational methods in noncoding RNA research. *Journal of Mathematical Biology*, 56(1):15—-49, 2008. 9

[51] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Byers. Big data: The next frontier for innovation, competition, and productivity. *Technical report, McKinsey Global Institute*, 2011. 23

[52] E. T. Matsubara. O Algoritmo de Aprendizado Semi-Supervisionado CO - TRAINING e sua Aplicação na Rotulação de Documentos. Master's thesis, Universidade de São Paulo, Butantã, São Paulo - State of São Paulo, 03178-200, 5 2004. 24

[53] A. Y. Matsukuma. Sequenciamento e anotação de parte do genoma de xylella fastidiosa. Master's thesis, Universidade de São Paulo, Instituto de Química-Dept. Bioquímica, USP - Av. Prof. Lineu Prestes, 748 - Vila Universitaria, São Paulo - SP, 05508-000, 9 2001. 2

[54] J. S Mattick. Non-coding RNAs: the architects of eukaryotic complexity. *EMBO Reports*, 2(11):986–991, 2001. 14

[55] T. M. Mitchell. Machine Learning. *McGraw-Hill ScienceEngineeringMath,*, 1997. 23, 27

[56] V. A. Moran, R. J. Perera, and A. M. Khalil. Emerging functional and mechanistic paradigms of mammalian long non-coding RNAs. *Nucleic Acids Research*, 40(14):6391–6400, 1986. 15, 16

[57] E. P. Nawrocki, D. L. Kolbe, and S. R. Eddy. Infernal 1.0: inference of RNA alignments. Technical report, HHMI Janelia Farm Research Campus, 19700 Helix Drive Ashburn VA 20147, 2008. 20

[58] NCBI. ORF Finder. https://www.ncbi.nlm.nih.gov/orffinder/, 2016. [Online; accessed 14-nov-2016]. 41

[59] NVO. Dogma central da Biologia Molecular. http://www.nvo.com/jin/nss-folder/scrapbookcell/central%20dogma%20.jpg, 2016. [Online; accessed 08-may-2016]. 12

[60] L. S. Ochi, C. R. Dias, and S. S. F. Soares. *Clusterização em Mineração de Dados*. Instituto de Computação – Universidade Federal Fluminense (IC – UFF), Av. Gal. Milton Tavares de Souza,, s/n - São Domingos, Niterói - RJ, 24210-346, 2016. Programa de Pós Graduação em Computação. 25

[61] Joint Genome Institute United States Department of Energy. Genome OnLine Database (GOLD). https://gold.jgi.doe.gov/, 2016. [Online; accessed 19-nov-2016]. 1

[62] J. V. A. Oliveira. *Identificação de snoRNAs usando Aprendizagem de Máquina*. Universidade de Brasília - Instituto de Ciências Exatas - Departamento de Ciência da Computação, Campus Universitário Darcy Ribeiro, Brasília - DF, 70910-900, 2 2014. Graduação em Ciências da Computação. 8

[63] Perl. Perl. https://www.perl.org/, 2016. [Online; accessed 15-oct-2016]. 41

[64] C. Ponting, P. Oliver, and e W. Reik. Evolution and functions of long noncoding RNAs. *Cell*, 136(4):629—-641, 2009. 15, 16

[65] Python. Python. https://www.python.org/, 2016. [Online; accessed 08-may-2016]. 43

[66] J. R. Quinlan. Induction of decision trees. *MACH. LEARN*, 1:81–106, 1986. 35

[67] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993. 34

[68] E. M. Real. Investigação de algoritmos sequenciais de agrupamento com pré-processamento de dados em aprendizado de máquina. *Faculdade Campo Limpo Paulista*, page 21, 2014. 25

[69] H. Schneider. *Identificação de RNA não-codificador utilizando SVM*. PhD thesis, Departamento de Ciência da Computação. Universidade de Brasília, Campus Universitário Darcy Ribeiro, Brasília - DF, 70910-900, 2015. Qualificação para o doutorado em preparação. 17

[70] H. W. Schneider, T. Raiol, M. M. Brigido, M. E. M. T. Walter, and P. F. Stadler. *A machine learning method to predict longnon-coding RNAs in transcriptomes*. Department of Computer Science, University of Brasilia, ICC Central, Instituto de Ciências Exatas, Campus Universitario Darcy Ribeiro, Asa Norte, CEP: 70910-900, Brasilia, Brazil, 2016. Artigo submetido. 40, 42, 91, 92, 97

[71] J. C. Setubal and J. Meidanis. *Introduction to Computional Biology*. PSW publishing company, Boston, 1997. 2, 6, 9, 10, 13, 14

[72] M. M. Silva. Uma abordagem evolucionária para o aprendizado semi-supervisionado em máquinas de vetores de suporte. Master's thesis, Universidade Federal de Minas Gerais - PPGEE/UFMG, Av. Pres. Antônio Carlos, 6627 - Pampulha, Belo Horizonte - MG, 31270-901, 11 2008. Programa de Pós-Graduação em Engenharia Elétrica. 26

[73] N. P. Silva and L. E. C. Andrade. Noções básicas de biologia molecular. *Revista Brasileira de Reumatologia*, 19(6):83–94, 2001. 13

[74] Slideplayer. Gene. http://images.slideplayer.com.br/1/50886/slides/slide_24.jpg, 2016. [Online; accessed 08-may-2016]. 9

[75] Sobiologia. Citologia. http://www.sobiologia.com.br/conteudos/figuras/Citologia2/DNA10.jpg, 2016. [Online; accessed 08-may-2016]. 10

[76] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. *MIT Press, Cambridge, MA*, 1998. 27

[77] M. Szymanski, J. Barciszewski, and V. A. Erdman. *Noncoding RNAs: Molecular Biology and Molecular Medicine, chapter Riboregulators*. Springer, 2003. 14

[78] P. N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining.*, volume 1. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2005. 33

[79] H. Timmers and L. Tora. The spectacular landscape of chromatin and ncRNAs under the tico sunlight. *EMBO reports*, 11(3):147—-149, 2010. 2

[80] I. J. Tinoco and C. Bustamante. How RNA folds. *Journal of Molecular Biology*, 293(2):271—-281, 1999. 17

[81] E. Torarinsson, M. Sawera, J.H. Havgaard, M. Fredholm, and J. Gorodkin. Thousands of corresponding human and mouse genomic regions unalignable in primary sequence contain common RNA structure. *Genome Research*, 16(7):885–889, 2006. 15

[82] B. T. M. Trevelim. *Mobot-Learn: Aprendizado por Reforço utilizando políticas parciais e macroestados na navegação de robôs móveis.* Escola Politécnica da Universidade de São Paulo, Avenida Professor Luciano Gualberto, Travessa 3, 380 - Butantã, São Paulo - SP, 05508-010, 2010. Graduação em Engenharia de Computação. 26, 27

[83] I. Ulitsky and D. Bartel. lincRNAs: genomics, evolution, and mechanisms. *Cell*, 154(1):26–46, 2013. 28, 29

[84] V. N. Vapnik. *The Nature of Statistical Learning Theory.*, volume 2. Springer, New York, 1995. 28

[85] J. Venter *et. al.* The sequence of the human genome. *Science*, 291(5507):1304–1351, 2001. 1

[86] P-J. Volders, K. Helsens, X. Wang, B. Menten, L. Martens, K. Gevaert, J. Vandesompele, and P. Mestdagh. LNCipedia: a database for annotated human lncRNA transcript sequences and structures. *Nucleic Acids Research*, 41(Database-Issue):246–251, 2013. 22

[87] J. D. Watson and F. H. Crick. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, 171:737–738, 1953. 1, 8, 10

[88] Wikimedia. RNA Nucleobases. https://upload.wikimedia.org/wikipedia/commons/thumb/d/de/RNA-Nucleobases.svg/774px-RNA-Nucleobases.svg.png, 2016. [Online; accessed 08-may-2016]. 7

[89] J. Wu, D. Delneri, R. O'Keefe, and et al. Non-coding RNAs in Saccharomyces Cerevisiae: what is the function? *Biochemical Society Transactions,*, 40(4):907, 2012. 14

[90] W. Zhang M. Guo X. Liu, D. Li and Q. Zhan. Long non-coding RNA gadd7 interacts with TDP-43 and regulates Cdk6 mRNA decay. *The EMBO Journal*, 31(23):4415–4427, 2012. 16

[91] U. Ørom and R. Shiekhattar. Noncoding RNAs and enhancers: complications of a long-distance relationship. *Trends in Genetics*, 27(10):433–439, 2011. 15